Mending Fractured Texts. A heuristic procedure for correcting OCR data

Jens Bjerring-Hansen[1], Ross Deans Kristensen-McLachlan[2], Philip Diderichsen[1] and Dorte Haltrup Hansen[1]

[1] *University of Copenhagen, Denmark*
[2] *University of Aarhus, Denmark*

Abstract

In this paper we present an OCR correction pipeline for 19th century printed Danish fraktur (gothic/blackletter). The work has been carried out at the University of Copenhagen in relation to a research project involving digital explorations of a corpus of some 900 Danish and Norwegian novels from 1870 to 1899, totalling app. 65 million words. Roughly 25% of these novels are printed in the traditional fraktur font, which was almost totally dominating in the beginning of the 19th century. These texts are important culturally, since they represent mostly forgotten, popular novels, however they pose technical and methodological challenges in terms of processing the text from printed page to digital corpus. In order to provide the best possible material for digital literary analysis as well as more linguistic studies, we designed an OCR correction pipeline for the fraktur part of the corpus consisting of several different heuristic correction steps, with reference to a gold standard. The first step is a preprocessing step which takes care of obvious and unambiguous OCR errors. In the second step we align our primary OCR output candidate (the output from Tesseract using the Fraktur.traineddata pretrained OCR model) with several other OCR output candidates and perform selective correction with reference to these. Especially the Danish "æ" and "ø" characters can be successfully recovered with reference to the Danish, non-fraktur dan.traineddata Tesseract model. Finally, in the third step, we employ the SymSpell spell checker to perform spelling correction backed by a word form dictionary hand-crafted from various relevant sources. The pipeline reduces the word error rate by 7.6 percentage points from 10.5% (89.5% correctly recognized word forms) to 2.8% (97.2% correctly recognized word forms) - an improvement of almost 73%. The character error rate (CER) similarly decreased from 1.94% to 0.54%.

Keywords

19th century literature, fraktur, Optical Character Recognition, OCR correction, Tesseract

1. Introduction

The background for this paper is a research project at the University of Copenhagen, *Measuring Modernity: Literary and Social Change in Scandinavia 1870-1900* (*MeMo*).[2] On the basis of a corpus of the app. 900 Danish and Norwegian novels printed in Denmark 1870–99 and with the help of digital literary analysis, the project aims to investigate the mental reflections of societal change in literature of the so-called "modern breakthrough" of the latter part of the 19th century, where Scandinavian societies underwent profound structural

The 6th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2022), March 15-18, Uppsala, Sweden.
EMAILS: jbh@hum.ku.dk (J. Bjerring-Hansen); rdkm@cas.au.dk (R. D. Kristensen-McLachlan); cpd@hum.ku.dk (P. Diderichsen); dorteh@hum.ku.dk (D. H. Hansen)

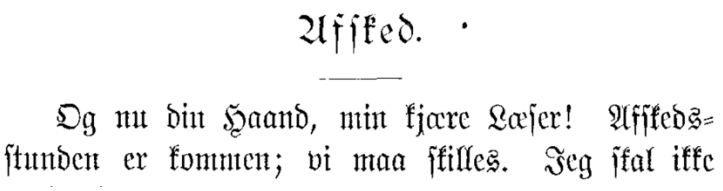


CEUR Workshop Proceedings (CEUR-WS.org)

[2] See https://nors.ku.dk/english/research/projects/measuring-modernity

changes, encompassing several, interlocking areas: demography, infrastructure, morals, culture, etc.

Within book production and print culture as such, a discrete, but important change took place in this era: the transition from gothic letters to the roman alphabet as default typeface for Dano-Norwegian books. For centuries the gothic letters, particularly the so-called "Fraktur", which was the dominating of the four typefaces in the Gothic family, the others were: Rotunda, Schwabacher and Textura [1] [2], had represented the norm in Danish language books, while roman letters or "Antiqua" where reserved to texts in Latin or other Roman languages as well as to individual foreign or loan words within a Danish, Fraktur context [3] [4]. Around 1850 only 5% of the yearly book production was set in Antiqua. In the 1870s, however, as a result of ambitions of international standardization and cultural modernization, instigated notably by the leading Copenhagen publishing house, Gyldendal, as well as the liberal daily press, the roman letters began to dominate, while the traditional typeface became associated with cultural parochialism and backwardness [4] [5].



Afſked.

Og nu din Haand, min kjære Læſer! Afſkeds-
ſtunden er kommen; vi maa ſkilles. Jeg ſkal ikke

Figure 1: Fraktur. Sample from the anounymous novel *Lars* from (as late as) 1895.

Corresponding with the general trend, with Antiqua on the rise and with Fraktur decreasing, but by no means eradicated, in our corpus of novels from 1870–99 roughly 20% of the texts are set in Fraktur. With the project's sociological ambitions of encompassing all novels from the period, not least the un-read and non-canonical ones, the subcorpus of Fraktur novels becomes crucial evidence. It seems that Fraktur lived on especially in popular novelistic genres, such as romances and historical novel, which constitute important parts of literary culture at large.

In other words, relating to a concept from the French sociologist Pierre Bourdieu, the question: Roman or Antiqua? covers an important cultural "distinction", giving evidence to systems of taste and prestige within the literary field of the late 19th century [6]. Thus, in digital analyses, our metadata category "Typeface" translates into a meaningful variable, which can be used for a segmentation of time series of literary trends.

However, in order to broaden the scope of our investigation of the literature of the period along these lines, it is of outmost importance that the quality of the subcorpus of Fraktur texts is on par with the quality of the Antiqua texts, which make up the major part of our corpus.

2. Related work

Optical character recognition is still a challenge in non-trivial cases such as archives of historical text, and OCR output may in many such cases benefit from post-OCR correction. [7] serves as an overview of state of the art techniques, and contains many valuable methodological recommendations.

In two recent competitions on post-OCR correction [8][9], the majority of the participating teams used machine learning (ML)/deep learning (DL) approaches. The best performers in both error detection and error correction were found among these. ML/DL approaches are of course prevalent nowadays [10] [11] [12] [13], but since our group currently does not have the resources to take this approach, it was encouraging that there

were also simpler, dictionary-based approaches among the good performers, achieving e.g. a 13% character error rate (CER) reduction for English monographs and a 23% CER reduction for French monographs (with relatively low initial CERs of 1-4%).

Some do post-OCR correction with humans in the loop [14] [15], achieving good results like a reduction in word error rate (WER) from 7.6% to 1.3%, given a human intervention on 2.2% of corpus tokens. A process like this would however still be quite resource-demanding in our case (a corpus in the tens of millions of tokens), so we decided to go for an automatic spelling correction approach.

Classical statistical and/or rule-based approaches employing spelling correction are able to improve word correctness from eighty-some to ninety-some percent, yielding one-digit WERs [16] [17] [18].

[18] reports one such method in a domain similar to ours: post-OCR correction of the archive of the British Medical Journal going back to 1840. They used the open source Hunspell spell checker with a few customizations: preprocessing of the corpus with a select few fixed pattern replacement rules; corpus frequency-based reranking of suggestions and selective correction; and augmentation of the Hunspell dictionary with a dictionary of medical terms. These modifications yield WER reductions between 7.5 and 16 percentage points (from 86.6 to 94.1 and from 70.2 to 86.2 percent correct words, respectively).
This type of results inspired us to pursue an approach along the same lines: a heuristic method based on spelling correction. In the next section, we detail our correction/evaluation pipeline.

3. Correction pipeline

The correction pipeline was built up iteratively to eventually comprise three main correction steps: 1) Replacement of safe error patterns, 2) Context-dependent correction with reference to alternative OCR sources, and 3) SymSpell spelling correction. The pipeline was built in Python.

The very first step was to determine which OCR source to use as a starting point. Four sources were available to us: ABBYY FineReader OCR scans (exact setup unknown) from the Danish Royal Library (Dan.: Det Kongelige Bibliotek, henceforth KB), and our own Tesseract 4.0[3] OCR scans using the pretrained models Fraktur.traineddata and frk.traineddata (both for non-Danish fraktur), and dan.traineddata (for Danish non-fraktur) [19].
We evaluated these four OCR sources against a gold standard consisting of one page of hand-corrected text from each of 60 fraktur novels (15760 running words in total).
Fraktur.traineddata had the best baseline quality with a word error rate of 10.46% (i.e. 89.54% correctly recognized words), and we thus decided to base the correction pipeline on this OCR source. (Results for the other sources: KB OCR WER: 13.97 (86.03% correct); frk.traineddata WER: 14.45 (85.55% correct); dan.traineddata WER: 29.54 (70.46% correct)).

By inspecting the errors in the OCR output from Fraktur.traineddata, we determined that a number of errors could safely be corrected by pattern substitution. They were all incorrect recognitions of the Danish letter "æ", and thus 'œæ', 'æœ', 'œe', 'eœ', and 'œ' could all simply be replaced by "æ".

During this work, we noticed that the different OCR sources had systematically different error types. In particular, the (non-fraktur-based) dan.traineddata output seemed to be correct in quite a lot of cases involving the letters "æ" and "ø", which are unknown to the non-Danish fraktur models. So we decided to see if this aspect of the otherwise quite bad

[3] https://github.com/tesseract-ocr/tesseract

dan.traineddata output could be utilized for selective correction. We ended up including a selective correction step that utilized information from the other OCR sources in the following way.

The starting point for the selective corrections is the lightly corrected Fraktur.traineddata output from the previous step. Given a substitution rule, say, 'o' => 'ø', a Fraktur token will be corrected if and only if a specific other OCR source (in this case: the dan.traineddata output) has a corresponding token with "ø" in one or more positions where the Fraktur token has "o". For instance, in the incorrect Fraktur token *lovlos* (for *lovløs* 'lawless'), the 'o' => 'ø' rule would apply to the second "o" if the dan.traineddata had the token *lovløs* because of the "ø" in position 5. The rule would even apply in the context of unrelated errors as in "louløs", "lovløf", or "louløf", as long as the "ø" is present in the correct position. If several replacements are relevant, these are all applied. E.g. a rule like 't' => 'k' will apply to *tysteste* (for *kyskeste* 'chastest, most chaste') to form the correct *kyskeste* in the context of a reference token like 'kyfkefte' due to the "k"s in position 1 and 4 corresponding to "t"s in the input. Thus, the s'es in the original input (underlined) would remain untouched, while a subset of the t's (italicized) would be replaced with reference to the alternative OCR: *t*ys*t*es<u>t</u>e.

As a final step, we applied the SymSpell spelling error correction algorithm in an attempt to correct some of the remaining errors.[4] The most important factor for SymSpell's correction quality is the quality of the frequency dictionary employed. The more representative of the input text, the better. We hand-crafted a frequency dictionary through a series of iterations, and eventually arrived at the following composition. The basis for our dictionary was a word form list generated from the Danish reference dictionary Ordbog over det danske Sprog (ODS).[5] Words of a length shorter than 4 characters were removed in order to remove a lot of esoteric short word forms. This word list was updated with frequency information (and missing word forms) generated from a collection of scholarly edited 19th century novels from the Danish *Arkiv for Dansk Litteratur* "Archive for Danish Literature",[6] plus a scholarly edited collection of Danish literary critic Georg Brandes' complete works.[7] Word forms on the ODS list not present in these sources were assigned a frequency of 1. Then, the list was augmented with frequencies from a 19m word collection of raw OCR-processed novels (in both Antiqua and Fraktur) from the period. The assumption here was that within an archive, a given word will be recognised correctly by the OCR software more frequently than misrecognised (cf. [18]). After some experimentation, it turned out that only words in the lower end of the frequency spectrum yielded any improvement. So, words within the collection with frequencies of between 3 and 42 were substituted in (their frequencies were normalized to the frequencies already on the list).

Finally, all correctly spelled person names were extracted from the gold standard texts and added with a relatively high, arbitrary frequency (if not already present). Names are obviously relevant for a spelling dictionary but not usually to be found in regular dictionaries. We extracted the names by searching for quotatives like 'said', 'answered', etc.; searching for uppercase initials would be useless since all nouns were uppercased in 19th century Danish. The resulting dictionary was lowercased, and the Danish character "å" replaced with the period equivalent "aa" (since "Å"/"å" was not introduced into Danish orthography until 1948).

The pipeline was run on one page of uncorrected Fraktur.traineddata output from each of 60 novels, and then compared to a gold standard consisting of the same 60 pages corrected by hand. We present the results in the next section.

[4] https://github.com/wolfgarbe/SymSpell, see Python package symspellpy.
[5] https://ordnet.dk/ods
[6] https://tekster.kb.dk/adl
[7] https://georgbrandes.dk

4. Results

The OCR correction pipeline described above was designed to produce a corpus file suitable for the Corpus Workbench system.[8] The corpus file contains a multitude of annotation layers with token-level error categorizations and error statistics for the various OCR sources – thus simultaneously serving as the source of a searchable and explorable OCR error corpus and as a dataset from which to extract relevant statistics (overall word error rates, etc.).[9] Here is a summary of the annotations:

- Overall annotations: **word**: Basic token layer; hand-corrected token (gold standard). **lineword**: Consecutive number of word on page line. **line**: Consecutive number of line on book page. **page**: Book page number. **novel_id**: Unique identifier of relevant novel. **sentword**: Consecutive number of word in sentence. **lemma**: Lemma of word. **pos**: Part of speech of word. **gold_infreq**: Whether word is in the frequency dictionary used by SymSpell.
- Annotations for each OCR source ("Fraktur", "dan", "frk", "kb", and last, but not least, "corr", i.e., the automatically corrected output of the pipeline, yielding 5 x 7 = 35 additional annotation layers):
    - **ocrtok**: output token of the given OCR source.
    - **leven**: Levenshtein edit distance from gold standard word.
    - **ratio**: Levenshtein ratio (word length corrected edit distance; between 0 and 1).
    - **cer**: Character error rate (implemented as 1 - ratio).
    - **levcat**: Classification of errors into categories such as 'match' (no difference), 'lev_1' (Lev. dist. 1), and 'split_lev_1' (Lev. dist. 1 with spaces involved).
    - **subst**: Substitutions; a representation of errors such as 'ø=o' (meaning: "correct 'ø' was erroneously replaced by 'o' in the OCR").
    - **infreq**: Whether ocrtok is in the frequency dictionary used by SymSpell.

These annotations/token categorizations were used during development to inspect the error types of the various OCR sources, and thus to inform improvements of the pipeline. In the end, the annotations were used to calculate the following results[10].

Table 1 below shows the overall effect of the three correction steps of the pipeline.

Table 1
Overall word error rate and correctness rate for various correction steps.

Output	Word error rate (%)	Correct words (%)
Fraktur.traineddata output without corrections	10.46	89.54

[8] https://cwb.sourceforge.io
[9] The corpus is currently available for search through the corpus interface Korp: https://alf.hum.ku.dk/korp/?mode=memo_frakturgold
[10] The gold standard, as well as other OCR text set in Antiqua and Fractur, was used in both development and testing of the pipeline, which can be methodologically problematic. However, we as humans are unlikely to overfit to our 'training set' nearly as much as a learning algorithm would. Thus, as our procedure is driven by heuristics based almost exclusively on domain knowledge, this gives us some confidence that our results will generalize well to the corpus at hand.

Fraktur + safe corrections	8.59	91.41
Fraktur + safe + selective corrections	4.14	95.86
Fraktur + safe + selective + SymSpell corrections (without names)	3.35	96.65
Fraktur + safe + selective + SymSpell corrections (with names)	2.84	97.16

The safe substitution of a small number of "æ" errors (cf. above) accounts for an initial improvement of 1.87 percentage points. The selective correction step accounts for a remarkable 4.45 additional percentage points, and the SymSpell step for another 1.3 percentage points - when names found in the data are added to the frequency dictionary. Our selective correction step is thus by far the most effective step in the pipeline, which achieves an overall improvement of 7.6 percentage points, or almost 73%.

In terms of character error rate. (CER), the overall results were as follows: The CER of the uncorrected output was quite low to begin with at 1.94%. The corrected output had a CER of 0.54% (an improvement of 72%). The smallest single-novel improvement was 0.56 percentage points, the largest 2.55 percentage points.

In terms of true and false positives and negatives, the results were as follows:

- 1271 (8.1%) true positives (i.e. actual errors successfully corrected)
- 14041 (89.1%) true negatives (i.e. correct words successfully skipped)
- 70 (0.4%) false positives (i.e. correct words erroneously 'corrected' to form new errors)
- 378 (2.4%) false negatives (i.e. actual errors missed (221 (1.4%)) or erroneously 'corrected' (157 (1%)))

These values add up to a precision of 0.95 (proportion of successful corrections out of all corrections), a recall of 0.77 (proportion of errors successfully corrected), and an F1 of 0.85 (harmonic mean of precision and recall).

When inspecting the 2.84% errors (448 of the 15760 running words in our material), we find that 84 (19%) of them are tokens that are simply lost in the OCR process.

Another 191 errors (43%) represent misinterpretations of single characters. Of these, 85 were misinterpretations of "æ", "ø", "Æ", or "Ø". Only a single "æ" was erroneously introduced - as a misinterpretation of "ø" (*kjære* ('dear') for *kjøre* 'drive'). The most frequent single character misinterpretations are:

Table 2
The most frequent single character misinterpretations.

x=y ('x becomes y')	N instances	Examples
ø=o	24	*møder* 'meets' => *moder* ('matures' (archaic)) *kørt* 'driven' => *kort* ('short')
æ=a	15	*være* 'be' => *vare* ('last, endure') *Mænd* 'men' => *Mand* ('man')
æ=e	9	*Hjærte* 'heart' => *Hjerte* *bortfjærnet* 'removed' => *bortfjernet*
H=E	8	(Only:) *Hr* (short for *Herre*) 'mister' => *Er* ('is')
i=j	7	*Kaptainen* 'the captain' => *Kaptajnen*

		Øienforblændelse 'hallucination' => *Øjenforblændelse*
Ø=O	7	*Ørkener* ('deserts') => *Orkener* *Ønsket* ('the wish') => *Onsket*
Ø=D	7	*Øine* ('eyes') => *Dine* ('yours') *Den* ('it') => *Øen* ('the island')

Notably, several of these errors seem to have to do with orthographic variation giving rise to lacunae and biases in the frequency dictionary:

- The printed form *Hjærte* as well as the substituted *Hjerte* 'heart' are both in the dictionary, but *Hjærte* is only half as frequent as *Hjerte*. (*Hjærte* is the only officially correct form in 1872, cf. the Danish spelling dictionary from 1872. By 1892 it is *Hjerte*[11]).
- Similarly, the printed form *bortfjærnet* 'removed' is not in the dictionary, whereas *bortfjernet* is. (*Fjærne* is the only correct spelling in 1872. By 1892 it is *fjerne*).
- The printed form Kaptainen is not in the dictionary, whereas Kaptajnen is. (The official spelling in 1872 is *Kaptejn* or *Kapitajn*, and in 1892 it is *Kaptajn*).
- The printed form *Øienforblændelse* 'eye-dazzlement, hallucination' is not in the dictionary, whereas *Øjenforblændelse* is. (In 1872 and 1892, *Öje* is the official spelling of 'eye').

49 errors (11%) represent insertions or deletions of 1 character, some of which also have to do with orthographic variation (e.g. *blanktpoleret* for *blankpoleret* 'brightly polished', where only *blanktpoleret* is in the dictionary); and the remaining 124 (28%) consist of more complex misinterpretations, with æ=ce, æ=oe, and, æ=ee among the most frequent ones.

Finally, it is worth mentioning that the quality of the individual novel (page)s of course varies. The original Fraktur output varies from 81.1% correct words (WER 18.9%) in the worst novel excerpt to 94.0% (WER 6%) in the best. The respective corrected output varies from 93.9% correct tokens (WER 6.1%) to 99.4%. (WER 0.6%).

5. Discussion

The results suggest that a substantial improvement of our OCR corpus is feasible, with only a few percent erroneous words remaining after correction. The improvement is on a par with similar approaches. [18] reports an improvement from 86.6% to 94.1% correct words for 1860s material; 82.6% to 90.5% for 1880s material; and 70.2% to 86.2% for 1900s material. (It should be noted that theirs are worst case measures; they are quite similar to the improvement of our worst novel excerpt from 81.1% to 93.9% correct words). [16] show a similar reduction of WER from 14.7% to 5.9% (85.3% to 94.1% correct words) (albeit only for words consisting of nothing but letters). And [17], exploiting Google's spelling suggestions, show error rate reductions from 21.4% to 3.1% (78.6% to 96.9% correct words) for English and from 12.5% to 3.1% (87.5% to 96.9% correct words) for Arabic.

The key element of our pipeline is the selective correction step informed by alternative OCR sources. This idea is not new. [20] use a voting algorithm to select candidates from two different OCR outputs generated from the same material. [15] observe that "the errors made by a primary OCR engine are highly correlated with its disagreements

[11] See https://rohist.dsn.dk

with subsequent secondary OCR engines", and exploit this to reduce errors prior to human assessment. [10] also use multiple alternative OCR outputs in their unsupervised deep learning approach, achieving a WER reduction from 41.8% to 23.4%. Our approach is informed by manual inspection of errors leading to selective, OCR source-specific correction rules. Their effectiveness owes to disagreements between OCR models being predictable, cf. the quotation above, yielding few spurious corrections when the rules are employed.

Like [18] we see significant improvement by preprocessing the texts with a select few context-free substitution patterns. These initial, safe error corrections facilitate the work of the subsequent steps.

When the text finally reaches the automatic spelling correction step, most of the errors have already been corrected. Nevertheless, this step still manages to correct almost one third (31%) of the remaining errors.

Additional experimentation has showed that additional simple, context-free substitution rules applied to the output of the pipeline (post-processing rules, as it were) would yield another 0.1% improvement - an improvement clearly in the realm of deminishing returns.

Manually adding names to the spell checker improved accuracy by a good 0.5 percentage points. Others too have observed the benefits of named entity recognition - and of adding other specialized vocabulary, e.g. medical terms [18] [21]. In our case, additional improvements might be gained from adding more types of named entitites other than just person names.

Systematically handling spelling variation might yield further, slight improvements, as suggested by the examples above. This could take the form of topic- or maybe novel-sensitive frequency based approaches along the lines of [22].

Some errors, however, almost one fifth (19%) of our remaining errors, represent tokens that are simply lost in the OCR process due to irregularities in the page images being OCR processed. These errors are out of reach for even the most sophisticated automatic correction procedures and rather require more careful image scanning, cropping, quantization, etc. - and clean, error-free print, of course.

Our results can hardly get any better without human intervention, which would however immediately increase the cost of correction considerably. It seems careful development of heuristic post-OCR correction procedures is still a viable method for lower-resourced projects (or languages) without an abundance of training data and other machine learning resources.

6. Future work

The next step for the project is to run the whole Fraktur part of the corpus through the correction pipeline as well as deploying it on the rest of the novels of the corpus, which are set in Antiqua. After that, the harder task of post-OCR cleaning a newspaper corpus from the same period (1870-1899) awaits. This is a more complex task due to the much more complex layout of newspaper pages. For this task, it may be necessary to rely to some extent on human intervention. The system described in [14] is currently under active development at the Copenhagen City Archive, and efforts are being made elsewhere in Scandinavia, too [21] [23], [24], affording a sense of optimism about this part of the project.

7. Conclusion

We have presented a promising method for post-OCR-correcting in the order of tens of millions of Fraktur tokens quite satisfactorily using a heuristic mix of hand-crafted rules and automatic spelling correction. We were thus able to reduce the amount of OCR errors in our

test corpus from 10.46% to 2.84%, a reduction of almost 73%. This shows that it is still possible to achieve good post-processing results without machine learning on the one hand and labour-intensive hand-correction on the other, which is good news for lesser-resourced projects and/or languages. In a cultural perspective the mending of the Fraktur texts is critical to paying due respect to important, but mostly forgotten parts of literary history.

8. Acknowledgements

This work was funded by The Carlsberg Foundation with support from the Digital Literacy initiative at Aarhus University. The authors would like to thank our anonymous reviewers for valuable feedback and pointers.

9. References

[1] Falk, V.: Bokstavsformer & typsnitt genom tiderna, 2nd. ed. Ordfront, Stockholm, 1989.

[2] Kap, A.: Fraktur. Form und Geschichte der gebrochenen Schriften, Schmitt, Mainz, 1993.

[3] Paulli, R.: Den sejrende antikva, Grafisk Cirkel, Copenhagen, 1940.

[4] Bjerring-Hansen, J.: Fraktur eller antikva? Om tekstens materialitet, typografisk usikkerhed i 1700-tallet og skriftstriden omkring 1800, in: Lychnos (2010), pp. 163–177.

[5] Rem T.: Materielle variasjoner. Overgangen fra fraktur til antikva i Norge, in: M. Malm *et al* (eds.), Bokens materialitet. Bokhistoria och bibliografi, Svenska Vitterhetssamfundet, Stockholm, 2009, pp. 151–173.

[6] Bourdieu, P.: Distinction: A Social Critique of the Judgement of Taste, Cambridge, Mass., Harvard University Press, 1984 [French orig. 1979].

[7] Nguyen, T. T. H., Jatowt, A., Coustaty, M., & Doucet, A. Survey of post-ocr processing approaches. ACM Computing Surveys (CSUR), 54(6), 1–37 (2021).

[8] Chiron, G., Doucet, A., el Coustaty, M., Moreux, J.-P.: ICDAR2017 Competition on Post-OCR Text Correction. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, France, pp.1423–1428 (2017).

[9] Rigaud, Christophe, Doucet, A., Coustaty, M., & Moreux, J.-P.: ICDAR2019 Competition on Post-OCR Text Correction. In: 15th International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, pp.1588–1593 (2019).

[10] Dong R., Smith. D. A.: Multi-Input Attention for Unsupervised OCR Correction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 2363–2372 (2018).

[11] Nastase,V., Hitschler, J.; Correction of OCR Word Segmentation Errors in Articles from the ACL Collection through Neural Machine Translation Methods. In: Proceedings of the The International Conference on Language Resources and Evaluation (LREC), pp. 706–711 (2018).

[12] Lyu, L., Koutraki, M., Krickl, M., Fetahu, B.: Neural OCR Post-Hoc Correction of Historical Corpora. In: Transactions of the Association for Computational Linguistics, 9: 479–493 (2021). DOI: https://doi.org/10.1162/tacl_a_00379

[13] Mokhtar, K., Bukhari, S., Dengel, A.: OCR Error Correction: State-of-the-art vs An NMT Based Approach. In: Proceedings of the 13th IAPR International Workshop on Document Analysis Systems, pp. 429–434 (2018).

[14] Richter, C., Wickes, M., Beser, D., Marcus, M.: Low-resource Post Processing of Noisy OCR Output for Historical Corpus Digitisation. In: In: Proceedings of the The International Conference on Language Resources and Evaluation (LREC), pp. 2331–2339 (2018).

[15] Abdulkader, A., Casey M.R.: Low Cost Correction of OCR Errors Using Learning in a Multi-Engine Environment. In: Proceedings of the 10th International Conference on Document Analysis and Recognition, pp. 576–580 (2009).

[16] Tong, X., Evans, D. A.: A Statistical Approach to Automatic OCR Error Correction in Context. In: 4th Workshop on Very Large Corpora, pp. 88–100 (1996).

[17] Bassil, Y., Alwani, M.: OCR Post-Processing Error Correction Algorithm Using Google's Online Spelling Correction. In: Journal of Emerging Trends in Computing and Information Sciences, 3:1 (2012).

[18] Thompson, P., McNaught, J., Ananiadou, S.: Customised OCR Correction for Historical Medical Text. In: Digital Heritage International Congress (2015). DOI: 10.1109/DigitalHeritage.2015.7413829

[19] Smith, R.: An Overview of the Tesseract OCR. In: Ninth International Conference on Document Analysis and Recognition (ICDAR) (2007). DOI: 10.1109/ICDAR.2007.4376991

[20] Klein, S. T., Popel, M.: A Voting System for Automatic OCR Correction. In: Proceedings of the SIGIR Workshop on Information Retrieval and OCR, pp. 1–19 (2002).

[21] Dannélls, D., Johansson, T., Björk, L.: Evaluation and refinement of an enhanced OCR process for mass digitization. In: Proceedings of the Digital Humanities in the Nordic Countries 4th Conference (DHN), pp. 112–123 (2019).

[22] Michael L. Wick, Michael G. Ross, Erik G. Learned-Miller Context-Sensitive Error Correction: Using Topic Models to Improve OCR. In Ninth International Conference on Document Analysis and Recognition (ICDAR) (2007).

[23] Kimmo Kettunen, K., Koistinen, M.: Open Source Tesseract in Re-OCR of Finnish Fraktur from 19th and Early 20th Century Newspapers and Journals – Collected Notes on Quality Improvement. In: Proceedings of the Digital Humanities in the Nordic Countries 4th Conference (DHN) (2019).

[24] Drobac, S., & Lindén, K. Optical Character Recognition with Neural Networks and Post-correction with Finite State Methods. International Journal on Document Analysis and Recognition (IJDAR), 23(4), 279-295 (2020).