Digitised Historical Text: Does it have to be mediOCRe?

Bea Alex, Claire Grover, Ewan Klein and **Richard Tobin**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, EH8 9AB, UK
{balex|grover|ewan|richard}@inf.ed.ac.uk

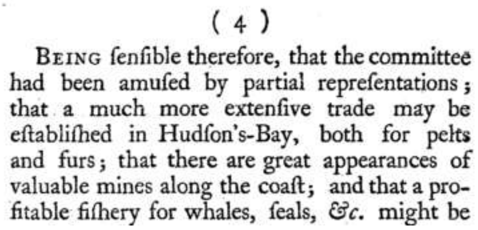
Abstract

This paper reports on experiments to improve the Optical Character Recognition (OCR) quality of historical text as a preliminary step in text mining. We analyse the quality of OCRed text compared to a gold standard and show how it can be improved by performing two automatic correction steps. We also demonstrate the impact this can have on named entity recognition in a preliminary extrinsic evaluation. This work was performed as part of the TRADING CONSEQUENCES project which is focussed on text mining of historical documents for the study of nineteenth century trade in the British Empire.

1 Introduction

The task of applying text mining techniques to digitised historical text faces numerous hurdles. One of the most troublesome of these is the 'garbled' nature of the plain text which often results when the scanned original undergoes Optical Character Recognition (OCR). In this paper we discuss two areas which cause problems: soft-hyphen splitting of word tokens and "long *s*"-to-*f* confusion. We evaluate the extent to which both issues degrade the accuracy of OCRed text compared to all OCR errors and describe methods for automatically correcting them.

A representative example of scanned text is shown in Figure 1, followed by the plain text output from OCR and the manually corrected gold standard text.



(4)
BEING ſenſible therefore, that the committee
had been amuſed by partial repreſentations;
that a much more extenſive trade may be
eſtabliſhed in Hudſon's-Bay, both for pelts
and furs; that there are great appearances of
valuable mines along the coaſt; and that a pro-
fitable fiſhery for whales, ſeals, &c. might be

Example 1: Fragment of scanned page from Robson (1752)[1]

```
( 4 ) BEING fenfible therefore, that the
committee had been amufed by partial
reprefentations ; that a much more
extenfive trade may be efnablifhed in
Hudfon's-Bay, both forpelts and furs;
that there are great appearances of
valuable mines along the coaft; and
that a pro-. fitable fifhery for whales,
feals, &c. might be
```

OCR output

```
( 4 ) BEING sensible therefore, that the
committee had been amused by partial
representations ; that a much more
extensive trade may be established in
Hudson's-Bay, both for pelts and furs;
that there are great appearances of
valuable mines along the coast; and
that a profitable fishery for whales,
seals, &c. might be
```

manually corrected output

As can be seen in Example 1, non-final lowercase letter *s* appears as "long *s*" (ſ or *ſ*). Not surprisingly, OCR tends to confuse this with

[1] http://eco.canadiana.ca/view/oocihm.20155/18?r=0&s=1

the lowercase letter *f*, since the only distinction between the two letters is that the long *s* has a nub on the left side of the letter whereas the real lowercase *f* has a nub on both sides. However, we need to correct this conflation of *s* with *f* in order to achieve reasonable accuracy in text mining.

A second issue can be seen in line 8 of the OCR output, where a line-break hyphen from the input text persists as a within-word hyphen, e.g., as `pro-fitable` (abstracting away from the additional insertion of a period and a space). Although the OCR has correctly recognised this 'soft' hyphen, it is still desirable to remove it in order to increase text mining accuracy. This removal can be regarded as a normalisation rather than correction *per se*.

The background and related work to our research are described in Sections 2 and 3, respectively. We have developed two tools which tackle the two issues mentioned above as accurately as possible; note that not every hyphen can be deleted and not every *f* should be turned into *s*. In both cases, we use a lexicon-based approach which is explained in more detail in Section 5. We evaluate the tools against a human corrected and normalised gold standard described in Section 4 in an attempt to quantify OCR accuracy and improvement. We describe the evaluation metric in Section 6 and report all the experiments we have conducted in Section 7. We include a preliminary extrinsic evaluation to determine the effect of text accuracy on named entity recognition.

2 Background

The Trading Consequences project aims to assist environmental historians in understanding the economic and environmental consequences of commodity trading during the nineteenth century. We are applying text mining to large quantities of historical text, converting unstructured textual information into structured data that will in turn populate a relational database. Prior historical research into commodity flows has focused on a small number of widely traded natural resources. By contrast, this project will provide historians with data from large corpora of digitised documents, thereby enabling them to analyse a broader range of commodities.

We analyse textual data from major British and Canadian datasets, most importantly the House of Commons Parliamentary Papers (HCPP),[2] the Canadiana.org data archive,[3] the Foreign and Commonwealth Office Collection from JSTOR[4] and a number of relevant books. Together these sources amount to millions of pages of text. The datasets include a wide range of official records from the British and Canadian governments, making them ideal for historical text mining. However, there are significant challenges in the initial step of transforming these document collections into a format that is suitable for subsequent text mining. Poor OCR quality is a major factor, together with artefacts introduced by the scanning process and nineteenth century language. For much of the corpus, the OCR was carried out several years ago, and is far inferior to what can be achieved nowadays with contemporary scanning hardware and OCR technology. The problems of OCR are aggravated for our corpus by the use of old fonts, poor print and paper quality, and nineteenth century language.

The project's underlying text mining tools are built on the LT-XML2[5] and LT-TTT2[6] tools. While they are robust and achieve state-of-the-art results for modern digital newspaper text, their output for historical text will necessarily involve errors. Apart from OCR imperfections, the data is not continuous running text but passages interspersed with page breaks, page numbers and headers and occasionally hand-written notations in page margins. In order for our text mining tools to pull out the maximum amount of information, we are carrying out automatic correction of the text as a preliminary processing step in our text mining pipeline.

[2] `http://parlipapers.chadwyck.co.uk/home.do`
[3] `http://www.canadiana.ca`
[4] `http://www.jstor.org/`
[5] `http://www.ltg.ed.ac.uk/software/ltxml2`
[6] `http://www.ltg.ed.ac.uk/software/lt-ttt2`

3 Related Work

Previous research on OCR post-correction includes use of a noisy channel model, where the true sequence of characters is generated given the noisy OCR output (Kolak and Resnik, 2002). Other studies have focussed on combining the output of multiple OCR systems to improve the text through voting (Klein and Kopel, 2002), efficient text alignments combined with dictionary lookup (Lund and Ringger, 2009) and merging outputs by means of a language model (Volk et al., 2011) or by active learning of human post-editing (Abdulkader and Casey, 2009).

Recent work has suggested improving OCR by making use of online search engine spelling corrections (Bassil and Alwani, 2012). While this has shown substantial reductions in the OCR error rates for English and Arabic text, the evaluation data sets are small with only 126 and 64 words, respectively. Combining dictionary lookup and querying candidates as part of trigrams in a search engine was also proposed by Ringlstetter et al. (2005), specifically to correct alphabet confusion errors in mixed-alphabet documents.

With specific focus on processing historical documents, there have been recent initiatives to improve OCR quality through manual correction via user collaboration. For example, the Australian Newspaper Digitisation Program set up an experiment to let the public correct the OCR output of historical documents, but found it difficult to measure the quality of the corrected text (Holley, 2009a; Holley, 2009b). The IMPACT project (Neudecker and Tzadok, 2010) is aimed at developing tools to improve OCR results via crowd sourcing to improve the digitisation of historical printed text. Related to this is the ongoing TEXTUS project[7] which plans to develop an open source platform for users to read and collaborate around publicly available texts. One of its planned functionalities is a mechanism that enables scholars to transcribe plain text versions of scanned documents.

The more specific issue of "long *s*" to *f* conversion has been addressed in an interesting but uncredited blog post.[8] This shows that a simple rule-based algorithm that maps *f*-containing strings not in the dictionary to words that are listed in the dictionary, improves OCRed text sufficiently to in turn improve recognition of taxon entities using TaxonFinder.[9] There is no detailed information on the dictionaries used, but we assume that they include general English dictionaries as well as specialised taxon dictionaries given that such terms are in Latin.

The second specific issue, namely fixing end-of-line hyphenation, has been addressed by Torget et al. (2011), who delete hyphens automatically in contexts s_1-s_2 whenever s_2 is not in the dictionary while the string with the hyphen omitted, namely s_1s_2, is in the dictionary. They do not provide information on how well their method performs.

There has also been work on determining the effect of OCR accuracy on text processing, be it information retrieval (IR) or text mining. In terms of IR, direct access to historical documents is hindered through language change and historical words have to be associated with their modern variants in order to improve recall. Hauser et al. (2007), for example, designed special fuzzy matching strategies to relate modern language keywords with old variants in German documents from the Early New High German period. Gotscharek et al. (2011) argue that such matching procedures need to be used in combination with specially constructed historical lexica in order to improve recall in IR. Reynaert (2008) proposes a language-independent method to clean high-frequency words in historical text by gathering typographical variants within a given Levenshtein distance combined with text-induced filtering.

OCR errors have been shown to have a negative effect on natural language processing in general. Lopresti (2005; 2008a; 2008b), for example, examines the effect that varying degrees of OCR accuracy have on sentence boundary detection, tokenisation and part-of-

[7] http://textusproject.org/

[8] http://inthefaith.net/rdp/botanicus/

[9] http://taxonfinder.sourceforge.net/

speech tagging, all steps which are typically carried out as early stages of text mining. Kolak and Resnik (2005) carried out an extrinsic evaluation of OCR post-processing for machine translation from Spanish into English and show that translation quality increases after post-correcting the OCRed text.

4 Data Preparation

We limited our analysis to the Early Canadiana Online collections since they contain a reasonably large number (∼83k) of different documents and since we wanted to get an idea of their OCR quality. We randomly selected a set of records from Canadiana, where a *record* is the OCRed counterpart of a page in the source document. More specifically, we shuffled the list of all Canadiana documents and randomly selected a record from the first 1,000 documents. These records were drawn from the first 20 records in the document; this limitation was due to the fact that Canadiana only provides free access to the first few scanned pages of a document, and our annotator needed to be able to access these in order to correct records and create a gold standard. When selecting a random record per document, we imposed some further restrictions, namely that the record had to be marked as English (`lang="eng"`) and it had to be more than 150 characters long. The latter length limitation was simply applied to avoid lots of short titles since we were mostly interested in running text. We also programmatically excluded records resembling tables of content or technical notes and disclaimers. This resulted in a set of passages which we gave to the human annotator to correct. For Experiment 1, described in Section 7.1, we used subsets of this random set: the original OCRed records (from now on referred to as ORIGINAL), the same records corrected and annotated as a gold standard (GOLD) and finally the records containing automatically corrected OCRed text (SYSTEM).

4.1 Preparing a Gold Standard

We asked the annotator to spend one day correcting and normalising records in ORIGINAL to create readable English text, leaving historical variants as they are but removing soft hyphens at the end of lines and changing wrongly recognised *f* letters into *s* as well as correcting other OCR errors in the text to the best of their ability. The annotator commented that while some records were relatively easy to correct, most required looking at the scanned image to determine what was intended. We asked the annotator to ignore any records which contained text in other languages, i.e., where the `lang` attribute in the original data was assigned wrongly. The annotator also ignored records which were so garbled that it would be quicker to write the record from scratch rather than correct the OCR output.

In total, the annotator corrected 25 records from ORIGINAL. These contained 8,322 word tokens, reduced to 7,801 in GOLD (including punctuation), when tokenised with our inhouse English tokeniser. An illustration of the annotator's input and output was shown in Section 1.

While "long *s*" confusion was found to be an issue in only four of the 25 records, the soft hyphen splitting issue was pervasive throughout the ORIGINAL dataset.

4.2 Preparing System Output

Finally, we processed the uncorrected versions of the records that comprised the GOLD set, using our tools to automatically remove soft hyphens and to convert incorrect *f* letters to *s* before tokenising the text to create the SYSTEM set. The methods of both these steps are described in detail in the next section, while their performance is evaluated in Section 7.

5 Automatic OCR Post-Correction

Our automatic OCR post-correction and normalisation involves two steps:

1. Removing end-of-line hyphens if they are soft hyphens; i.e., they hyphenate words that would normally not be hyphenated in other contexts.

2. Correcting *f* letters to *s* if they were a "long *s*" in the original document.

5.1 End-of-line Soft Hyphen Deletion

The program for removing end-of-line soft hyphens is called `lxdehyphen`. It expects an XML file containing elements corresponding to lines of text. If all lines of text in a page are concatenated into one in the OCR output, then the text is first split at tokens ending in hyphen + whitespace + new token by inserting an artificial newline character. The program then tokenises the text simply using whitespace and newline as delimiters to identify hyphenated words; a more sophisticated tokenisation is not needed for this step. If the last token on a line ends with a hyphen it is considered as a candidate for joining with the first token of the next line. The tokens are joined if, after removing the hyphen and concatenating them, the result is either a word that appears in the Unix/Linux system dictionary `dict`,[10] or is a word that appears elsewhere in the document. The latter heuristic, which can be considered as using the document itself as a dictionary, is very effective for documents with technical terms and names that do not appear in `dict` as well as for historical documents which contain historical variants of modern terms. Provided that a word appears somewhere else in the document unhyphenated, it will be recognised and the soft hyphen will be removed.

The tokenisation markup is then deleted and a more sophisticated tokenisation can be carried out. If soft hyphen splits have been removed, this will typically result in a reduction in the number of tokens.

5.2 *f*→*s* Character Conversion

The crucial component of the *f*→*s* conversion tool is `fix-spelling`, an `lxtransduce` grammar that replaces words based on a lexicon that maps misspelled words to the correct version. Lexicons can be constructed for various purposes; in this case, we use the lexicon `f-to-s.lex` for correcting poorly OCRed historical documents where the "long *s*" has been conflated with *f*. The `f-to-s.lex` lexicon is created from a corpus of correct text. For each word in that corpus, a word frequency distribution is collected and all the possible misspellings caused by the long-*s*-to-*f* confusion are generated. It is possible that some of these generated words will also be real words (e.g., *fat* < *sat*).[11]

The unigram frequency counts of each word in the corpus is therefore used to determine its likelihood. For example, *difclose* will be corrected to *disclose* because *difclose* does not occur in the corpus. *fat* will be corrected to *sat* because *sat* occurs more often. But *feed* will not be changed to *seed* because *feed* occurs more often. The corpus can be chosen to be similar to the target texts so that the results are more reliable; in particular, using old texts will prevent words that were not common then from being incorrectly used. In our experiment, we used the text from a number of books in the Gutenberg Project as the corpus to create the lexicon.[12]

6 Text Alignment and Evaluation

In the set of experiments described below, we compare different versions of a given text to determine a measure of text accuracy. In order to carry out the evaluation, we first need to align two files of text and then calculate their differences. OCR software suppliers tend to define the quality of their tools in terms of character accuracy. Unfortunately, such figures can be misleading if the digitised text is used in an information retrieval system, where it is searched, or if it is processed by a text mining tool. An OCRed word token that contains only a single character error (i.e., insertion, deletion or substitution) will score more highly on this measure than one which contains multiple such errors. However, the word will still be incorrect when it

[10] The dictionary (`/usr/share/dict/words`) can vary between operating systems. We ran our experiments using the Scientific Linux release 6.2 with a dictionary containing 479,829 entries.

[11] We use $w_1 < w_2$ to indicate that string w_1 has been derived from w_2, either by one of our tools, as in this case, or by OCR.

[12] The books were *Adventures of Sherlock Holmes*, *Christmas Carol*, *Dracula*, *Great Expectations*, *Hound of the Baskervilles*, *Paradise Lost*, *Tale of Two Cities*, *The Adventures of Sherlock Holmes*, *The History of England, vol. 1* and *Works of Edgar Allen Poe, vol. 1.*

comes to analysing and processing it automatically, especially as many text processing tools involve dictionary, gazetteer or other exact-match word-lookup methods.

In this paper, therefore, we evaluate the quality of the converted text in terms of word error rate (WER):

$$\text{WER} = \frac{I + D + S}{N}$$

where I is the number of insertions, D the number of deletions and S the number of substitutions between the hypothesis and the reference string, while N is the number of word tokens in the reference (e.g., the GOLD dataset). We calculate WER by means of the GNU `wdiff` program,[13] a front end to `diff`[14] for comparing files on a word-per-word basis, where a word is anything between whitespace or newline characters. `diff` aligns a sequence of text by determining the longest common subsequence (Miller and Myers, 1985; Myers, 1986; Ukkonen, 1985). `wdiff` then determines the counts for the words in common between both files as well as word insertions, deletions and substitutions. Note that `wdiff` counts a word as a substitution if it is replaced or is part of a larger replacement. This is not an issue for our evaluation as we are not interested in the distribution of insertions, deletions and substitutions but merely their sum.

7 Experiments

The following first three experiments report on the quality of Canadiana's OCRed text, and the extent to which our processing tools can improve it. The forth experiment provides an initial examination of how the OCR quality affects the recognition of commodity entity mentions, the latter being one of the aims of the TRADING CONSEQUENCES project.

7.1 Experiment 1

In Experiment 1, we first compare the difference between the ORIGINAL and the GOLD data set. The difference between these two versions of the text shows how much the original OCRed text needs to change to be transformed to running English text without errors. We then run both the `lxdehyphen` and the $f \rightarrow s$ conversion step over the original text and create the SYSTEM$_{all}$ data set. By comparing the SYSTEM$_{all}$ against GOLD and determining their differences, we can then calculate the improvement we have made to the ORIGINAL data in light of all the changes that could be made.

Test Set	I	D	S	WER
ORIGINAL	71	24	1,650	0.224
SYSTEM$_{all}$	73	24	1,434	0.196
SYSTEM$_{dehyph}$	74	24	1,519	0.207
SYSTEM$_{f2s}$	70	24	1,567	0.213

Table 1: Number of insertions (I), deletions (D) and substitutions (S) and word error rate (WER) when comparing the different test sets against the GOLD dataset of 7,801 tokens.

Table 1 shows that the ORIGINAL OCRed text has a word error rate of 0.224 compared to the corrected and normalised GOLD version. After running the two automatic conversion steps over the ORIGINAL data set creating SYSTEM$_{all}$, word error rate was reduced by 12.5% to 0.196. Given that the GOLD set was created by correcting all the errors in the OCR, this is a substantial improvement.

We also ran each correction step separately to see how they each improve the OCR individually. The scores for SYSTEM$_{dehyph}$ and SYSTEM$_{f2s}$ show that the soft hyphen removal step contributed to reducing the error rate by 7.6% (0.017) to 0.207 and $f \rightarrow s$ conversion by 4.9% (0.011) to 0.213. Soft hyphens are an issue throughout the text but long-s-to-f confusion only occurred in four out of the 25 records, which explains why the effect of the latter step is smaller. The results show that after fixing the OCR for two specific problems automatically a large percentage of errors (87.5%) still remain in the text.

7.2 Experiment 2

Since we evaluate each conversion tool in light of all other corrections of the OCRed text, it is

[13] http://www.gnu.org/software/wdiff/

[14] http://www.gnu.org/software/diffutils/

quite difficult to get an idea of how accurate they are. We therefore evaluated each tool separately on a data set which was only corrected for their particular issue. For Experiment 2, we therefore hand-corrected the 25 Canadiana records again, but this time only for soft hyphen deletion, ignoring all other issues with the OCR. We call this data set GOLD$_{dehyph}$ and compare it to ORIGINAL to determine the difference between both texts. We also compare GOLD$_{dehyph}$ to SYSTEM$_{dehyph}$ created in the previous experiment.

Test Set	I	D	S	WER
ORIGINAL	1	0	177	0.022
SYSTEM$_{dehyph}$	1	0	47	0.006

Table 2: Number of insertions (I), deletions (D) and substitutions (S) and word error rate (WER) when comparing against GOLD$_{dehyph}$ (8,203 tokens).

The results in Table 2 show that soft hyphens in the OCRed text reduce the word token accuracy by 0.022 compared to the normalised gold standard. The automatic fixing reduces this error by 72.7% to 0.006. Error analysis showed that the remaining differences between SYSTEM$_{dehyph}$ and GOLD$_{dehyph}$ are caused by missing soft hyphen deletion because the split tokens in question contained other OCR problems and were either not present in the dictionary or not repeated in the text itself, e.g., *3ndow- ment* (< *endowment*) or *patron- aye* (< *patronage*). The `lxdehyphen` tool correctly did not remove hyphens which were meant to stay in the text.

7.3 Experiment 3

As with Experiment 2, we wanted to separately evaluate the performance of the *f*→*s* conversion step. Since the GOLD data set only contained four page records with the "long *s*" confusion, we created a bigger gold standard of ten Canadiana page records which all have the issue in their original OCR and which the annotator hand-corrected only for *f*→*s* ignoring all other errors in the text. We first compare the original text of this data set (f2s_ORIGINAL) to the hand-corrected one (f2s_GOLD) to quantify the effect of the problem on the text. We then ran the automatic *f*→*s* conversion over the f2s_ORIGINAL set and compared the output (f2s_SYSTEM$_{f2s}$) to the gold standard (f2s_GOLD).

Test Set	I	D	S	WER
f2s_ORIGINAL	0	0	720	0.095
f2s_SYSTEM$_{f2s}$	0	0	206	0.027

Table 3: Number of insertions (I), deletions (D) and substitutions (S) and word error rate (WER) when comparing against f2s_GOLD (7,618 tokens).

The results in Table 3 show that the character confusion problem increases the word error rate dramatically by 0.0945. This means that OCRed documents in which this issue occurs will benefit significantly from an automatic post-correction step that fixes this problem. Our own system output reduces the error by 71.6%, yielding a word error rate of 0.027. An error analysis of the remaining differences shows that, as with the previous experiment, the tool does not successfully convert false *f* letters in tokens containing other OCR issues, e.g., *adminiftzr* (< *administer*) or *fireet* (< *street*). In a few cases, it also causes some real-word errors, i.e., converting *f*→*s* when it should not have done so. This happens either for tokens where a letter other than *s* was confused with *f* during the OCR process, e.g., *fhe* (< *the*) is converted to *she*, or for tokens whose unigram frequency is smaller than that of the converted token, e.g., *fees* was incorrectly converted to *sees*.

7.4 Experiment 4

After the annotator finished correcting all of the 25 GOLD page records, she was asked to mark up commodity entities in that data set. Our definition of a commodity entity is something that is sold or traded and can be either natural or man-made. The annotator marked up 167 entities including general commodities such as *apples* and *copper* but also more specific items such as *Double Ended Taper Saw Files* or *Iron Girders*. We then transferred the annotations manually to the ORIGINAL and the SYSTEM$_{all}$ data as accurately as possible.

Our next task in the TRADING CONSEQUENCES project is to develop a named entity recogniser which identifies commodities that were important in nineteenth century trade. This work is being carried out in collaboration with historians at The University of York, Toronto, and involves creating a commodities thesaurus/ontology using the SKOS framework.[15] This thesaurus will be the basis for our further system development. In the meantime, we are using WordNet[16] to approximate commodity terms; that is, we use a chunker to recognise noun chunks in the text and label them as commodity mentions if they are a hyponym of the WordNet classes substance, physical matter, plant or animal.

Test Set	Found	Correct
ORIGINAL	79	28
GOLD	86	34
SYSTEM$_{all}$	80	29

Table 4: Number of found and correct commodity entities in the various test sets.

Even though this is a very crude method with a low performance, the effects of the OCR are apparent in the results shown in Table 4. The smallest number of entities and of correct entities are found in the uncorrected ORIGINAL set. 7 more entities (of which 6 are correct) are recognised in the completely corrected and normalised GOLD set. The two automatic correction steps which reduce the distance from ORIGINAL to GOLD by 12.2% lead to one additional correct entity being recognised compared to the original OCR. While this means that more OCR post-corrections would be desirable to improve the named entity recognition, we also believe that the effect of both existing post-correction tools will become more apparent as we improve our commodities recognition system.

8 Discussion and Conclusion

It is widely recognised that much of the OCRed text currently available for historical documents falls far short of what is required for accurate text processing or information retrieval. We have focussed on automatically fixing two issues in such text, namely soft hyphen deletion and *f*→*s* conversion. We have evaluated both methods and shown that together they deal with just over 12% of all word error problems in our sample. In addition, each of them successfully deals with around 72% of relevant cases. We have carried out an error analysis of where the tools fail to yield the correct results. Finally, we have described a very preliminary study which indicates that fixing and normalising the OCRed text is beneficial to named entity recognition.

As we have seen, even after these steps, a large number of OCR errors remain in the text. We will need to address at least some of these to achieve our desired level of text mining performance. As part of this task, we intend to explore whether some of the techniques we briefly reviewed in Section 3 can be combined with our current approach.

When creating the gold standard data, we found that the quality of the OCRed text from different sources can vary depending on factors such as the quality of the scan, the quality of original script and printing, the contents of a page, and so one. In a few cases the text we provided to the annotator turned out to be too garbled to understand. If humans are unable to correct such records then automatic systems will have little chance of doing so, and text mining will produce no or very useless information. We are therefore planning to integrate a further pre-processing step into our system which tries to estimate text accuracy and rejects documents from processing whose accuracy falls below a certain threshold.

Acknowledgments

We would like to thank our TRADING CONSEQUENCES project partners at The University of York, Toronto, The University of St.Andrews and EDINA for their feedback (Digging Into Data CIINN01). We would also like to thank Clare Llewellyn for her valuable input and help with the annotation.

[15] http://www.w3.org/2004/02/skos/

[16] http://wordnet.princeton.edu/

References

Ahmad Abdulkader and Mathew R. Casey. 2009. Low cost correction of OCR errors using learning in a multi-engine environment. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, ICDAR '09, pages 576–580, Washington, DC. IEEE Computer Society.

Youssef Bassil and Mohammad Alwani. 2012. OCR post-processing error correction algorithm using Google's online spelling suggestion. *Journal of Emerging Trends in Computing and Information Sciences*, 3(1):90–99, January.

Annette Gotscharek, Ulrich Reffle, Christoph Ringlstetter, Klaus U. Schulz, and Andreas Neumann. 2011. Towards information retrieval on historical document collections: the role of matching procedures and special lexica. *IJDAR*, 14(2):159–171.

Andreas Hauser, Markus Heller, Elisabeth Leiss, Klaus U. Schulz, and Christiane Wanzeck. 2007. Information access to historical documents from the Early New High German period. In L. Burnard, M. Dobreva, N. Fuhr, and A. Lüdeling, editors, *Digital Historical Corpora- Architecture, Annotation, and Retrieval*, Dagstuhl, Germany.

Rose Holley. 2009a. How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine*, 15(3/4).

Rose Holley. 2009b. Many hands make light work: Public collaborative OCR text correction in Australian historic newspapers. National Library of Australia, Technical Report.

Shmuel T. Klein and Miri Kopel. 2002. A voting system for automatic OCR correction. In *Proceedings of the Workshop On Information Retrieval and OCR at SIGIR*, pages 1–21.

Okan Kolak and Philip Resnik. 2002. OCR error correction using a noisy channel model. In *Proceedings of the second international conference on Human Language Technology Research*, pages 257–262.

Okan Kolak and Philip Resnik. 2005. OCR post-processing for low density languages. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 867–874.

Daniel Lopresti. 2005. Performance evaluation for text processing of noisy inputs. In *Proceedings of the Symposium on Applied Computing*, pages 759–763.

Daniel Lopresti. 2008a. Measuring the impact of character recognition errors on downstream text analysis. In B. A. Yanikoglu and K. Berkner, editors, *Document Recognition and Retrieval*, volume 6815. SPIE.

Daniel Lopresti. 2008b. Optical character recognition errors and their effects on natural language processing. In *Proceedings of the second workshop on Analytics for Noisy Unstructured Text Data*, pages 9–16.

William B. Lund and Eric K. Ringger. 2009. Improving optical character recognition through efficient multiple system alignment. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, JCDL'09, pages 231–240.

Webb Miller and Eugene W. Myers. 1985. A file comparison program. *Software - Practice and Experience*, 15(11):1025–1040.

Eugene W. Myers. 1986. An O(ND) difference algorithm and its variations. *Algorithmica*, 1:251–266.

Clemens Neudecker and Asaf Tzadok. 2010. User collaboration for improving access to historical texts. *LIBER Quarterly*, 20(1).

Martin Reynaert. 2008. Non-interactive OCR post-correction for giga-scale digitization projects. In *Proceedings of the 9th international conference on Computational Linguistics and Intelligent Text Processing*, pages 617–630.

Christoph Ringlstetter, Klaus U. Schulz, Stoyan Mihov, and Katerina Louka. 2005. The same is not the same—postcorrection of alphabet confusion errors in mixed-alphabet OCR recognition. In *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 406–410.

Joseph Robson. 1752. *An Account of Six Years Residence in Hudson's-Bay from 1733 to 1736, and 1744 To 1747.* Printed for J. Payne and J. Bouquet, London.

Andrew J. Torget, Rada Mihalcea, Jon Christensen, and Geoff McGhee. 2011. Mapping texts: Combining text-mining and geovisualization to unlock the research potential of historical newspapers. University of North Texas Digital Library, White Paper.

Esko Ukkonen. 1985. Algorithms for approximate string matching. *Information and Control*, 64(1-3):100–118.

Martin Volk, Lenz Furrer, and Rico Sennrich. 2011. Strategies for reducing and correcting OCR errors. In C. Sporleder, A. Bosch, and K. Zervanou, editors, *Language Technology for Cultural Heritage*, chapter 1, pages 3–22. Springer-Verlag, Berlin/Heidelberg.