Synergy of Nederlab and @PhilosTEI: diachronic and multilingual Text-Induced Corpus Clean-up

Martin Reynaert
TiCC, Tilburg University; CLST, Radboud University Nijmegen
reynaert@uvt.nl

Abstract

In two concurrent projects in the Netherlands we are further developing TICCL or Text-Induced Corpus Clean-up. In project Nederlab TICCL is set to work on diachronic Dutch text. To this end it has been equipped with the largest diachronic lexicon and a historical name list developed at the Institute for Dutch Lexicology or INL. In project @PhilosTEI TICCL will be set to work on a fair range of European languages. We present a new implementation in C++ of the system which itself has been tailored to be easily adaptable to different languages. We further revisit prior work on diachronic Portuguese in which it was compared to VARD2 (Baron, 2011) which had been manually adapted to Portuguese. This tested the new mechanisms for ranking correction candidates we have devised. We then move to evaluating the new TICCL port on a very large corpus of Dutch books known as EDBO, digitized by the Dutch National Library. The results show that TICCL scales to the largest corpus sizes and performs excellently raising the quality of the Gold Standard EDBO book by about 20% to 95% word accuracy. Simultaneous unsupervised post-correction of 10,000 digitized books is now a real option.

Keywords: OCR post-correction, multilingual, diachronic

1. Introduction

The aim of this paper is to show how Text-Induced Corpus Clean-up (TICCL) is currently deployed in two concurrent but unrelated projects in the Netherlands: Nederlab[1] and @PhilosTEI[2].

Nederlab is a five year NWO[3] project that ambitiously aims to make available online all existing corpora for Dutch in an environment that will not only allow for exploration of the texts but also for their selection, analysis and exploitation for research purposes.

@PhilosTEI is a one year CLARIN-NL[4] project in which an open-source workflow from digital image to TEI XML formatted digitized text is being built.

Both projects share that they deal with texts that have been digitized and therefore contain Optical Character Recognition (OCR) noise. It is our task to try and reduce the impact of this noise and to restore the electronic version of the original paper-bound text to the best possible extent.

Through the synergy of further development of TICCL in both projects, we may look forward to having available a multilingual OCR post-correction system fit for diachronical work.

2. Project Nederlab and TICCL

2.1. The Nederlab project

The Nederlab project aims to be a research portal to all digitized texts relevant to the Dutch national heritage, the history of Dutch language and culture (from about A.D. 800 to the present) offering one open access, user-friendly and tool-enriched web interface, to allow scholars to simultaneously search and analyze data from texts spanning the full recorded history of the Netherlands, its language and culture. Nederlab's added value is in creating a user-friendly infrastructure for researchers, aimed at promoting cooperation and synergy, and it is hoped, at the formulation of new, often interdisciplinary, research questions.

The route followed in Nederlab is to convert all the texts incorporated into a common format, FoLiA XML (van Gompel and Reynaert, 2013). In their turn all the research and analysis tools (will) have been adapted to this format. If already available online, the texts remain as they are at their original location and the linguistically or otherwise enriched versions link to these.

The focus in Nederlab is currently to incorporate the vast digital text collections of the Dutch National Library[5] (KB) as well as the contents of the Digitale Bibliotheek voor de Nederlandse Letteren[6] (DBNL - The Digital Library of Dutch Literature).

2.2. TICCL in Nederlab

Vast amounts of texts such as newspapers, magazines and books[7] have been digitized by means of OCR over the past years, e.g. at the Dutch National Library. If there is one thing all results of large digitization programmes have, it is that they are riddled with OCR misrecognition errors.

The effect of these errors commonly best recognized is that when searching these text collections, one has no idea of what one misses. The first aim of applying OCR post-correction to these collections is therefore to enhance the recall of text retrieval.

A second effect, recognition of which is an early result of the Nederlab project, is that if it is one's goal to further automatically enrich the corpora linguistically, the poor recognition accuracy rates of the various OCR engines have

[1] http://www.nederlab.nl/
[2] http://axiom.vu.nl/PhilosTEI.html
[3] http://www.nwo.nl/en
[4] http://www.clarin.nl/
[5] http://www.kb.nl/en
[6] http://www.dbnl.org/
[7] http://www.delpher.nl/

a direct and very noticeable effect on the tools one has at hand for further linguistic enrichment, be they tokenizers, POS-taggers, lemmatizers or Named Entity recognizers, to name but the ones in most common use. Not only will the tools perform less well in terms of performance accuracy achieved on the corpora, but one may also very well be confronted with drastic processing speed reductions. A slow tool needs not present an undue problem when one wants to merely process a single book, which is e.g. the aim of @PhilosTEI, it becomes another proposition altogether if, as we are doing in Nederlab, one wants to process hundreds of thousands of them, or e.g. millions of newspapers articles. To this latter condition we return in Section 7.3., where we discuss more in depth the subcorpus of almost 10,000 books we are currently focusing on in Nederlab.
To gain insight in how well TICCL in its new C++ version currently performs on diachronic Dutch text, we measure in Section 7.3. how well it performs on a 1789 book digitized by the KB.

3. Project @PhilosTEI and TICCL

3.1. TICCL in @PhilosTEI

The second project in which TICCL plays its role is far more modest in scope than Nederlab. Its aim is quite simple and straightforward. Philosophers today increasingly require electronic versions of the works they study. In CLARIN-NL project @PhilosTEI we are therefore building a work flow of web services which will allow individual researchers to upload digital images of the book's pages and receive back after processing a well formatted electronic text version fit for further building into e.g. a critical edition of the work. In the work flow, it is TICCL's task in its guise as the web service TICCLops, to enhance the text's quality, fully automatically.
This small project fits into a larger research programme[8], called 'Tarski's revolution'. The works studied in fact present a cross-section of European languages: German, Polish, French, Italian, etc.

3.2. The @PhilosTEI project

The envisaged combination of web services will consist of existing open-source tools that are individually turned into RESTful web applications/services by means of Clam[9]. The first webservice around Tesseract[10] will convert the digital text images into electronic text and pass its hOCR html-output on. The second webservice will consist of text format convertors in the guise of XSLTs from hOCR to TEI XML.
Again concurrently with the @PhilosTEI project, the KB is a partner in an international project, eMOP[11], which happens to be almost identical in part of its aims and even in some technical details of implementation. Both projects have the stated aim of using only available open-source tools. The projects are nevertheless nicely complementary: where @PhilosTEI has some budget for further developing fully automatic post-correction, eMOP has a more substantial budget and international partners to develop infrastructure for crowd-sourcing of the post-correction process.

4. TICCL: Corpora used and tasks performed

4.1. Portuguese: revisiting previous work on transcription of historical text

In (Reynaert et al., 2012) we compared VARD2 with our own tool TICCL. We concluded that given a better ranking mechanism TICCL might outperform VARD2 even if the latter had been specially and manually adapted and tuned to historical Portuguese.
We have since developed a better ranking mechanism for correction candidates proposed by TICCL and so revisit the same work. We here measure the extent to which our improved ranking helps performance on the task of transcribing historical Portuguese into its modern variant.
For the evaluation experiments of TICCL and VARD2 we use a subpart of 200 letters from the CARDS-FLY corpus. These letters were manually normalized by one linguist but difficult cases were discussed with a second expert. This data set was split in 100 letters for training and tuning the tools, and 100 letters were set apart as a true test set. The test set contains 37,372 tokens of which 6,978 (19%) are spelling variants that need to be detected and normalized by the tools.

4.2. Dutch: Martinet book

In an earlier CLARIN-NL demonstrator project called TICCLops[12] we built a web service and application which demonstrates how Text-Induced Corpus Clean-up works on the basis of an 18th century Dutch book, 'Kort begrip der waereldhistorie voor de jeugd' [E.: 'Brief insights into World History for the young'] by J.F. Martinet, printed in 1789.
This book, known as DPO35, was digitized by the KB and an OCR ground truth for the book was developed in the Impact project. We have now built an OCR post-correction gold standard for both the historical text and its derived modern version. More detailed information about the differences between OCR ground truths and gold standards in general and detailed statistics about DPO35 can be found in the companion paper to the current one (Reynaert, 2014).
TICCLops as first delivered in fact suffered from data sparseness when evaluated on this book: a/ the historical lexicon it was equipped with was in fact a word list published in 1914 (obtained via Gutenberg[13]), the word list contains few non-lemma word forms and represents considerably more modern Dutch than the language of Martinet. b/ this being a historical text in both the sense that it deals with history and that it is in itself pre-Darwinianly old, it abounds with archaic names – hardly ever encountered today except perhaps in the Bible or other religious texts – which are futhermore in an archaic spelling.

[8] http://axiom.vu.nl/
[9] http://proycon.github.io/clam/
[10] http://code.google.com/p/tesseract-ocr/
[11] http://www.kb.nl/en/research/emop-early-modern-ocr-project
[12] Demonstrator and user manual online at: http://ticclops.dev.inl.nl/
[13] http://www.gutenberg.org/files/22722/22722-h/woordenlijst1914-full.html

We now try to alleviate both shortcomings by incorporating what must be the largest extant historical lexicon for Dutch as well as its accompanying name list. Both were developed at INL[14], the Dutch Institute for Lexicology, partner in both Nederlab and @PhilosTEI. They were deliverables of the European project Impact and are available through the Impact Centre of Competence[15]. We here measure their effect on OCR post-correction of the Martinet book.

5. Better resources for diachronical Dutch

5.1. Better ranking of correction candidates

As described in (Reynaert, 2010) TICCL used to rank the correction candidates or CCs by subsequent sorting of the CC lists on the basis of some combination of their corpus frequencies, their Levenshtein distance (Levenshtein, 1966) or LD to the correction focus (CF), i.e. the word form being corrected, and the observed frequency of the character confusion between CC and CF. In fact, either of the three may in particular cases have to be given prominence over the others. We have therefore been searching for a more solid solution and finally arrived at this: for each of the sources of ranking information we sort the candidate lists separately. We then assign a separate numeric ranking to these lists, ascending, so e.g. the CC with the highest corpus frequency will be assigned rank 1, the next: 2, etc. We next sum the rankings for each CC and divide this by the sum total of all the rankings over all the CCs associated with the CF. Subtracting this from 1 gives us a ranking score between 0 and 1 for each CC.

For Portuguese, examining the ranked correction candidates, we soon realized that the evidence available for the ranking is in fact inadequate to sufficiently differentiate between the various CCs. As we explained in (Reynaert et al., 2012), the greater morphological variation in Portuguese as compared to Dutch and to an even greater extent to English results in a wider profusion of CCs differing e.g. only in the morphological suffix, which has an adverse effect on the ranking. Given that our new ranking scheme allows for easy incorporation of new features which may enhance the ranking, we have now incorporated word overlap features:

- word suffix overlap information (whether last two characters match or not)
- word onset overlap information (whether the first characters of the word match or not)
- common longest substring information (largest overlap in number of letters ranked first)

We finally added one more extra ranking feature, which is whether the character confusion between a word pair was also observed before, in the training set.

This indeed results in better ranking for the Portuguese transcription task, as we will show in the experimental results in Table 1.

5.2. INL historical lexicon and name list for Dutch

TICCL has now been equipped with some of the best lexical resources for diachronical Dutch. More information on the INL historical lexicon and name list for Dutch is to be found in (de Does and Depuydt, 2013).

6. A new TICCL[16]

6.1. Background

With the addition of the now more expensive ranking scheme, in part due to an expensive actual Longest Common Subsequence module, the TICCL Perl prototype proved to scale no longer to the sizes of the larger corpora we work on. TICCL was therefore gradually and modularly ported to C++.

All the C++ TICCL modules are distributable, i.e. they can be given as many computer processor threads as one has available. In the current tests, we gave them 30 threads to work on, except for TICCL-rank which proved still buggy in distributed mode and was therefore run in single-thread mode.

As implied by their name, some tools were developed to work on FoLiA XML. How we deal with OCRed text and spelling suggestions and corrections in this xml format is described more fully in (Reynaert, 2014).

In order to understand TICCL one must have a basic understanding of anagram hashing. This is described in depth in (Reynaert, 2010) which details the differences in the approaches to spelling correction represented by the focus word approach versus the character confusion approach. We here venture a brief informal explanation of how anagram hashing works.

In anagram hashing each character in the alphabet is assigned a large numerical value. This value should be chosen in such a way that each character stands at an exact Euclidean distance from each other character and even of all possible combinations of characters. For combinations of characters the values for the individual characters making up the string are summed. The absolute numerical difference between e.g. 'b' and 'c' will then be exactly the same as the numerical result of subtracting the anagram value for 'bat' from 'cat'. We used to derive the anagram value for each character from its particular code page value raised to the power 5. In this work the lowest anagram value we use is 100 raised to the fifth power.

We describe the individual C++ modules of TICCL in the next subsection.

6.2. TICCL C++ modules

Module 1: Lexstat: TICCL language adaptation TICCL can now be adapted to another language by deriving a language specific alphabet from a lexicon for the language. In this work we have used the expanded available open-source Aspell dictionary for Dutch (the idea being that TICCL can in this straightforward way be adapted to the many languages for which Aspell dictionaries have

[14] http://www.inl.nl/

[15] http://www.digitisation.eu/

[16] TICCL in its new C++ implementation is to be available via http://ticclops.uvt.nl/

been compiled). Aspell dictionaries come in a coded version detailing the possible morphological variants for the lemmata in the dictionary. It provides an expansion facility by way of which one may obtain all the expanded word forms. This expanded list we use as input for Lexstat. Its first output is a descending character frequency list for the language. This may then be 'clipped', i.e. characters due to loan words which have an observed frequency below a chosen threshold may be removed. The clipped alphabet is then used to set the anagram value for each character (clipped characters and unseen characters will then share a fixed anagram value, i.e. are in fact binned together to reduce the search space) and further to build the character confusion list and its attendant anagram values up to the Levenshtein distance envisaged. LD in this work was set at 2. Based on a character frequency cut-off of 20, given the Dutch expanded Aspell dictionary, the Dutch alphabet we work with here has 36 characters. We add a 'bin' value which represents all punctuation marks (except for the word-internal marks, the apostrophe and the hyphen, which were incorporated in the alphabet from the lexicon. We further add yet one more bin value which then represents all unseen characters (which may later occur in the corpus), the clipped ones and all digits. An alphabet of 38 characters amounts to 275,651 character confusion values given LD 2.

Module 2: FoLiA-stats The first step in running TICCL is taken by FoLiA-stats. As its name implies the program assumes FoLiA xml as its input. Its output is a word unigram frequency list for the corpus. This is merged with the lexicon, each word type in the validated lexicon being assigned a large frequency which if observed in the corpus is augmented with the corpus frequency. This provides a handle on what word forms are trusted and which are not.

Module 3: TICCL-unk In the next step, we run TICCL-unk. This program examines each word type in the frequency list and applies filters. Word types that are deemed unrecoverable by TICCL because they exceed a particular threshold of non-alphabetical characters are set aside in an output file. Word types that have any number of leading or trailing punctuation mark are likewise set aside in another output file, linked to their 'tokenized' variants. These latter and all the approved word types are written to a cleaned frequency file.

Module 4: TICCL-anahash TICCL-anahash uses the anagram values assigned to the characters in the alphabet file obtained by Lexstat to calculate the anagram values for all the word strings in the cleaned frequency file. Anagrams are chained to the anagram value they share.

Module 5: TICCL-indexer In the third step, the cleaned frequency file is converted to a corpus anagram hash on the basis of the character anagram values produced by Lexstat. TICCL-indexer is a purely numerical step. For each anagram value in the character confusion list it collects the values in the corpus hash that when summed with the particular character confusion list value give a hit on another value in the anagram hash. It thus builds an index to all word pairs displaying all the possible confusions given the LD limit. This means we perform an exhaustive lookup for all possible character confusions regardless of where they occur in the word forms represented by the corpus hash. The output of TICCL-indexer can be reconstructed as a full Spellnet in the terms of (Choudhury et al., 2007), i.e. a full graph of all the word forms in the lexicon and the corpus linked by an edit distance of at most the Levenshtein distance imposed. Within this work we have worked with LD = 2.

Module 6: TICCL-LDcalc TICCL-LDcalc has the heaviest task. On the basis of the TICCL-indexer output it actually retrieves the word pairs linked through exhibiting a particular character confusion. It evaluates all the actual word pairs composable from the chained anagrams linked to the anagram values in the corpus hash and discards the pairs that exceed the LD limit. It further discards all the pairs whose frequencies exceed the value set for validated lexicon words, thereby filtering out the bulk of the real-word pairs or confusables. At the same time it collects or computes the information we will use to rank the CCs.

Module 7: TICCL-rank TICCL-rank performs the current final step of actually ranking the CCs for each focus word. It outputs a list of focus word and CC pairs, with basic frequency information, the LD and the confidence score on which CCs are ranked. Optionally it outputs an extended list with all the ranking information for debugging purposes. This is humanly interpretable, which we find a comfortable fact.

Module 8: FoLiA-correct The final module actually edits the FoLiA xml files to be corrected on the basis of the output of TICCL-rank. It also substitutes by 'UNK' (for: UNKnown) the character strings that were deemed unrecoverable by TICCL-unk. These strings are replaced by 'UNK' in the TICCL copy of the original paragraph[17]. This is an exceedingly fast module which processes the EDBO in a matter of a few hours, but is likely the most memory intensive one. Nevertheless we have not seen its memory requirements exceeding 10GB in these tests.

6.3. New TICCL ranking

We have now incorporated more ranking features into TICCL than we experimented with in our Perl prototype. Some of these would not have been applicable or of any use in the Portuguese historical spelling normalization task, but prove of worth in OCR post-correction. We duly note in this respect the ranking on number of capitalized versions of a word form observed within the corpus.
TICCL currently has the following ranking features:

- frequency of the CC
- cumulated frequency of the various capitalized versions of the CC
- CC frequency as binned by order of magnitude
- LD

[17]We may resort to another designation for these strings as we recently found out in Named Entity annotation that Unk is in fact a real place name.

- An approximation of the length of the Longest Common Subsequences. Obtained by subtracting the LD from the longest of the two variants in terms of character frequencies.
- Whether the CC is part of the canonical lexicon or was only observed in the corpus
- Whether the first letter of focus and CC match
- Whether the last two letters of focus and CC match
- Whether the particular character confusion anagram value pertaining to the focus and CC pair was seen with a frequency above a particular threshold in the INL historical lexicon word form and lemmata pairs.
- the number of times the particular character confusion anagram value pertaining to the focus and CC pair was observed in the pairs retained by LDcalc.
- how many variously capitalized versions of the CC were observed.

The non-binary features are ranked depending on whether more rather than less is better. Higher frequencies are naturally better than lower, the top frequency is assigned rank 1, the second 2, and so on. Identical frequencies are assigned the same rank. Smaller LDs are better, LD = 1 is thus given rank 1. The binary features are assigned rank 1 if they are true, rank 2 when no true. Word forms in the canonical lexicon are boosted by rank 1, the others downgraded by being given rank 10.
The ranks obtained over all the features are summed and are then divided by the number of features at play. For each CC, the ranking score obtained is then divided by the sum of the ranking scores of all the CCs for the particular focus word. This result is subtracted from 1 in order to obtain a confidence score between 0 and 1. The CC with the highest confidence score is then ranked best first.

7. Evaluation

To gain an idea of TICCL's adaptability to other languages we in this paper study its current performance on historical Portuguese. The task here is automatic transcription of historical into contemporary Portuguese. In particular we measure the impact of better ranking of the correction candidates on Portuguese.
To study TICCL's fitness on the task of post-correcting diachronical texts, we measure the combined effects of better ranking and of adding a historical lexicon and name list on a Dutch historical book.

7.1. General remarks on the evaluations

We measure the performance of the tools and compute accuracy, recall, precision and their harmonic mean, F-score, (van Rijsbergen, 1975) on the spelling variants. Our own general and specific views on evaluating spelling and OCR post-correction are more fully developed in (Reynaert, 2008).

7.2. Performance on Portuguese

In Table 1 we present detailed performance results on the transcription of historical Portuguese on a test set of 100 letters from the CARDS-FLY corpus. Having equipped TICCL with a more informed ranking system for handling the CCs it retrieves, we can now say that our unsupervised system outperforms a purpose-trained VARD2. These improved results are primarily due to better ranking facilities implemented in the Perl prototype version of TICCL used in the prior work.

Tool	acc	prec	recall	f-score
VARD2	94.65	96.99	73.63	83.71
TICCL2	93.50	94.38	69.33	79.94
TICCLrank	95.68	96.38	79.84	87.33

Figure 1: Best-first ranked results on the test set of 100 CARDS-FLY corpus letters. VARD2 and TICCL2 results are from (Reynaert et al., 2012). TICCLrank gives the new results primarily due to better ranking facilities implemented in the Perl prototype version of TICCL used in the prior work.

7.3. Performance on Dutch

In this section we move from the mere few hundred letters of Portuguese text in the previous section to a digitized corpus of about 10,000 Dutch OCRed books, about 1.7 million pages of printed text. The corpus is known as EDBO or 'Early Dutch Books Online', DPO35 is part of this collection. We give some more statistics on the corpus in Table 2.

Unit	amount
Books	10K
Pages	1.7M
Tokens	435M
Types	20M

Figure 2: Statistics on the Dutch books in EDBO

While the new TICCL port still not has all the provisions incorporated in the prototype, e.g. absolute correction and some provisions for dealing with split and run-on words, we here nevertheless present results obtained on OCR post-correction on the for Dutch unprecedented scale of EDBO. We aim to establish whether TICCL scales, as well as to what extent it can perform the joint task of correcting OCR errors and normalizing diachronic text to the modern canonical spelling.
We here report on the very first tests to scale we have performed with the new TICCL. As a result, performance scores reported should be seen as preliminary and indicative only. The tests are meant to establish whether what we aim to do is feasible and whether the system truly scales to the magnitude of the task it is set. Further, the tests may well reveal unanticipated remaining shortcomings such as oversights or even bugs in the code.

We test the system on all the Dutch books in EDBO, simultaneously. The collection is freely available online as the 'basiscollectie' (E: basic collection) under the 'books' tab in the KB site Delpher. Statistics about the books are presented in Table 2. The collection spans publication years (about) 1700 to 1800, but the bulk of the books were published between 1780 and 1800. Our assumption is that if TICCL manages to process these books in a reasonable amount of time, using a reasonable amount of resources, it will be capable of handling any digitized text collection. The main reason why we should want to process so many books in one go is mainly a matter of lexical statistics: TICCL is firmly based on the assumption that larger digitized text collections provide valuable information to the correction process thereby facilitating better correction. The tests we perform were designed to illustrate and to stave the assumption. The main reason why TICCL is able to process a corpus of the current size is because anagram hashing affords an economical and therefore scalable solution to the string matching problems posed by the veritably huge search space represented by the EDBO corpus.

We evaluate on a single book from the collection, DPO35, for which we have built a historical and a modern gold standard for the text based on the OCR ground truth that was produced within the European project Impact. After Delpher came online in late November 2013 we soon found out that it in fact presents a newer digitized version of DPO35 than the one we had acquired directly from the KB in a previous project, TICCLops. This is unfortunate as we can now not direct the reader to the site to see for themselves in case we give examples of OCR misrecognition errors in the digitized version. We take it that the images presented there are the same as used for producing both OCR versions. The newer OCR version we will incorporate into our gold standard as soon as possible.

What we now have is a basic TICCL which we want to evaluate on its current merits. We set its reach in terms of edit distance to LD = 2. The anagram hash and frequency list at its disposal has word unigrams only, hence, we do not try to solve split words or run-ons. We think local word context, as might be provided by word n-grams but we do not have available, is required to properly resolve short word OCR problems. We therefore set TICCL to search for variants only on words from character length 6, inclusive, only, up to length 36 – rather arbitrarily.

We have now incorporated more ranking features into TICCL than we experimented with in our Perl prototype. Some of these would not have been applicable or of any use in the Portuguese historical spelling normalization task, but prove of worth in OCR post-correction, we duly note the ranking on number of capitalized versions of a word form observed within the corpus.

7.4. Evaluation results on EDBO

Table 3 gives an overview of the results obtained in the evaluation on EDBO of the new TICCL. We have run the system five times, using 4 different lexicons or combinations thereof. We have run it on two corpora, one which consists solely of the single book DPO35 we have both a historical and a modern gold standard for, and the other consisting of the about 10,000 Dutch books in EDBO. In the table we refer to the lexicons with the label 'L'. The first lexicon (1) is the one we have used over the last few years in prior work on Dutch which we discussed before in Section 4.2. The second lexicon (2) we have briefly discussed in Section 5.2. The third is the concatenation of 1 and 2. This was then merged with the INL historical names list to form (4).

Results are good and in line with expectance. Note that we give results on the full range of problems posed by the book, measuring TICCL's performance in terms of the full task of solving each and any problem, be it names split into individual characters by the typesetting ploy of the period to emphasize them by spacing out the characters or substitution errors in page numbers. It should be understood that these are problems TICCL can currently not solve, in part as a consequence of the limitations we imposed on it we discussed above.

L	C	acc	prec	recall	f-score
10 best-first ranked					
1	A	91.92	99.77	61.01	75.71
1	B	93.23	99.48	66.92	80.02
2	B	93.42	99.39	69.22	81.60
3	B	94.50	99.46	72.77	84.05
4	B	95.97	99.81	77.27	87.11
best-first ranked					
4	B	94.51	99.79	70.98	82.96

Figure 3: Evaluation results on the task of fully automatically normalizing and OCR post-correcting as measured on the full DPO35 Gold Standard. The results clearly show the effect of using different lexicons (L): the old TICCL lexicon (1), the INL historical Dutch lexicon (2), the previous two combined (3), the combined lexicons further enhanced with the INL Historical names list (4). We have measured on two corpora (C), i.e. the book DPO35 only (A), or the book as part of the 10,000 books EDBO collection (B). We first list results as measured on the 10 best first ranked CCs, then list the highest best-first ranked combination.

Both historical lexicon and name list have a very noticeable positive effect on TICCL's performance on the historical Dutch book. The accuracy of the OCR version of DPO35 compared to the modern text is 76.24%, to the historical text 88.94%. The accuracy of the TICCL corrected version is 95.97%. This is a highly encouraging result which makes us conclude that simultaneous OCR post-correction on the scale of 10,000 digitized books is now a distinct possibility. Running EDBO took our research server about 30 hours.

8. Discussion

The results show that it is likely that TICCL manages to improve recall on noisy historical texts by improving the overall quality of the texts as is evident from the improvement in overall accuracy of the text of the Portuguese letters and on the Dutch digitized historical book, DPO35.

However, much work remains to be done to fully automatically improve the quality of the texts to such an extent that

they are no longer likely to present major challenges to the tools one would like to deploy on them in order to linguistically annotate and enrich them.

It seems unlikely to us that OCR post-correction alone will ever succeed in achieving this. Especially in the huge collection of digitized newspapers Nederlab aims to process, we encounter such elevated amounts of word strings misrecognized to such an extent that no post-correction tool will ever be able to sufficiently undo the effects of the OCR 'noisy channel'.

An interesting venue for future research is to see whether TICCL output may serve to determine, on the basis of its not having been linked to any plausible CCs, whether a particular string, text line or other likely unit was originally misrecognized by the OCR process so that it should be sent back to be re-examined by perhaps a newer version of the commercial software used, or to one or more of the open-source alternatives.

We have demonstrated that anagram hashing allows for exhaustively filtering the search space represented by several million word strings from a large and noisy OCRed corpus up to all possible character confusions represented by a medium large European alphabet given LD 2. In so doing, we have gone up to two orders of magnitude higher in terms of character confusions explored than (Reffle and Ringlstetter, 2013) who we still think are hampered by the fact that the Finite State technology they use cannot scale in the same way. This may help to explain why they present a system with many nice features but which is geared to processing a single book at a time. Our results here show conclusively, at least to ourselves, that handling books within the context of a great many similar books is a far better proposition to help alleviate the problems caused by the transition from paper to electronic media given the actual numbers of digitized works being produced.

Based on the contribution of the INL Historical lexicon and name list we have seen here, we do not hold with the recommendations in (de Does and Depuydt, 2013) to tailor one's lexicon before attempting OCR post-correction by removing likely confusables in the form of infrequent words for their more frequent alikes. Apart from the practical problems involved given that we now use a more than 1 million word types lexicon, the fact remains that infrequent words do appear.

9. Concluding remarks

We have in this paper described ongoing work in two Dutch projects, Nederlab and @PhilosTEI. While the projects are in themselves unrelated, the synergy of both allows us to further develop our OCR post-correction system TICCL which leads to a multilingual system fit for diachronic work, as well as fit for distribution in open-source. We have demonstrated the multilinguality of TICCL by applying it to a Romance language, Portuguese, and to the Germanic language Dutch it has mainly been developed on. We have evaluated its capabilities on diachronic text for both languages and have measured in passing the performance gains brought about by better informed ranking of the transcription or correction candidates and by providing it with better lexicons.

10. Acknowledgements

It is with immense pleasure that I here acknowledge the great contribution made to this work by TiCC senior scientific programmer Ko van der Sloot. His expert programming now makes me feel confident enough to finally release TICCL into the world. I further gratefully acknowledge support from NWO in project Nederlab and from CLARIN-NL in projects @PhilosTEI (CLARIN-NL-12-006) and OpenSoNaR (CLARIN-NL-12-013).

11. References

Baron, A. (2011). *Dealing with spelling variation in Early Modern English texts*. Ph.D. thesis, University of Lancaster, Lancaster, UK.

Choudhury, M., Thomas, M., Mukherjee, A., Basu, A., and Ganguly, N. (2007). How difficult is it to develop a perfect spell-checker? A cross-linguistic analysis through complex network approach. *Proceedings of the second workshop on TextGraphs: Graph-based algorithms for natural language processing*, pages 81–88.

de Does, J. and Depuydt, K. (2013). Lexicon-supported OCR of eighteenth century Dutch books: a case study. In Zanibbi, R. and Coüasnon, B., editors, *DRR*, volume 8658 of *SPIE Proceedings*. SPIE.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710. Original in *Doklady Akademii Nauk SSSR* 163(4): 845–848 (1965).

Reffle, U. and Ringlstetter, C. (2013). Unsupervised profiling of OCRed historical documents. *Pattern Recognition*, 46(5):1346 – 1357.

Reynaert, M., Hendrickx, I., and Marquilhas, R. (2012). Historical spelling normalization. a comparison of two statistical methods: TICCL and VARD2. In Mambrini, F., Passarotti, M., and Sporleder, C., editors, *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities*, pages 87–98. Lisbon: Colibri.

Reynaert, M. (2008). All, and only, the errors: more complete and consistent spelling and OCR-error correction evaluation. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Reynaert, M. (2010). Character confusion versus focus word-based correction of spelling and OCR variants in corpora. *International Journal on Document Analysis and Recognition*, 14:173–187. 10.1007/s10032-010-0133-5.

Reynaert, M. (2014). On OCR ground truths and OCR post-correction gold standards, tools and formats. In *Proceedings of Datech 2014*. ACM.

van Gompel, M. and Reynaert, M. (2013). FoLiA: A practical XML Format for Linguistic Annotation - a descriptive and comparative study. *Computational Linguistics in the Netherlands Journal*, 3.

van Rijsbergen, C. J. (1975). *Information Retrieval*. Butterworths, London.