# Non-Interactive OCR Post-Correction for Giga-Scale Digitization Projects

Martin Reynaert

Induction of Linguistic Knowledge, Tilburg University, The Netherlands

**Abstract.** This paper proposes a non-interactive system for reducing the level of OCR-induced typographical variation in large text collections, contemporary and historical. Text-Induced Corpus Clean-up or TICCL (pronounce 'tickle') focuses on high-frequency words derived from the corpus to be cleaned and gathers all typographical variants for any particular focus word that lie within the predefined Levenshtein distance (henceforth: LD). Simple text-induced filtering techniques help to retain as many as possible of the true positives and to discard as many as possible of the false positives. TICCL has been evaluated on a contemporary OCR-ed Dutch text corpus and on a corpus of historical newspaper articles, whose OCR-quality is far lower and which is in an older Dutch spelling. Representative samples of typographical variants from both corpora have allowed us not only to properly evaluate our system, but also to draw effective conclusions towards the adaptation of the adopted correction mechanism to OCR-error resolution. The performance scores obtained up to LD 2 mean that the bulk of undesirable OCR-induced typographical variation present can fully automatically be removed.

## 1 Introduction

This paper reports on efforts to reduce the massive amounts of non-word word forms created by OCRing large collections of printed text in order to bring down the type-token ratios of the collections to the levels observed in contemporary 'born-digital' collections of text. We report on post-correction of OCR-errors in large corpora of the Cultural Heritage. On invitation by the National Library of The Netherlands (Koninklijke Bibliotheek - Den Haag) we have worked on contemporary and historical text collections. The contemporary collection comprises the published Acts of Parliament (1989-1995) of The Netherlands, referred to as 'Staten-Generaal Digitaal' (henceforth: SGD)[1]. The historical collection is referred to as 'Database Digital Daily Newspapers' (henceforth: DDD)[2], which comprises a selection of daily newspapers published between 1918 and 1946 in the Netherlands. The historical collection was written in the Dutch spelling 'De Vries-Te Winkel', which in 1954 was replaced by the more contemporary spelling

[1] URL: http://www.statengeneraaldigitaal.nl/

[2] URL: http://kranten.kb.nl/ In actual fact, this collection represents the result of a pilot project which is to be incorporated into the far more comprehensive DDD.

used in the SGD. Both collections should be seen as pilot projects for extensive digitization projects underway in which a large part of the newspaper collection present in the National Library will be made publicly available online in the course of the next few years. A nice consequence of the fact that both collections we have worked on here are already available online is that any example given in this paper can be independently verified. If we claim that the English word 'restoring' is in fact an OCR-misrecognition of the word 'regeering' (i.e. De Vries-Te Winkel spelling for the contemporary word 'regering' ('government'), any reader can look it up on the DDD website and see the actual word highlighted in yellow in the digital image from which the OCRed text version was created by a company, using the OCR-software Abbyy FineReader version 6.

In the next Section we discuss the nature of lexical errors in large text collections. In Section 3 we present previous work in relation to the aims of this paper and in Section 4 we introduce our approach based on anagram hashing. Section 5 deals with the evaluation. We conclude in Section 6.

## 2 OCR-Errors And Other Lexical Variation In Corpora

Commercially available Optical Character Recognition (OCR) systems boast high accuracy these days. Given that a system reaches 99% word accuracy, it should nevertheless not be lost from view that this in fact means that one word will have been misrecognized out of every hundred words processed.

Fig. 1 shows the Vocabulary Growth Curves (VGCs) [1] obtained with the zipfR package due to [2] for the three text collections we deal with in this paper. VGCs show how many new words are seen the more text is read. The attendant growth lines for hapax legomena are also plotted. TWC02 is a contemporary one-year newspaper corpus, covering the year 2002, composed mainly of 5 national Dutch newspapers. We use the corpus here as a kind of reference for the 'normal' vocabulary growth one would expect to see. Its curve at first shows a greater vocabulary growth than the SGD. This is easily explained by the fact that in newpapers far more topics are touched upon than in a typical debate in Parliament, which is likely centered around far fewer topics, calling upon a smaller vocabulary. At around 60M words the vocabulary growth of the TWC02 starts to slow down, and the SGD vocabulary continues to rise. We take this to be the effect of OCR-errors within the SGD. In sharp contrast to both these curves, the VGC of 'Het Volk', one of the newspapers in the DDD, exhibits a markedly sharp ascent. We list more statistics on the corpora in Table 1. Note the tremendous type-token ratio observed for 'Het Volk' (articles - 1918).

As concerns hapax legomena, words observed only once in a particular corpus, represented in the plot by the finer lines mimicking the lines representing the full vocabulary, the TWC02 displays a fairly normal rate of 44,2% overall. Around 50% of words in text are typically expected to be hapax legomena [3] (p.199). The SGD has a slightly higher rate at 55,2% overall, but again 'Het Volk' tops with a rate of hapaxes at 86,4% overall. This exceedingly high ratio of hapaxes to types in 'Het Volk' shows that OCR-errors, which are often considered to be

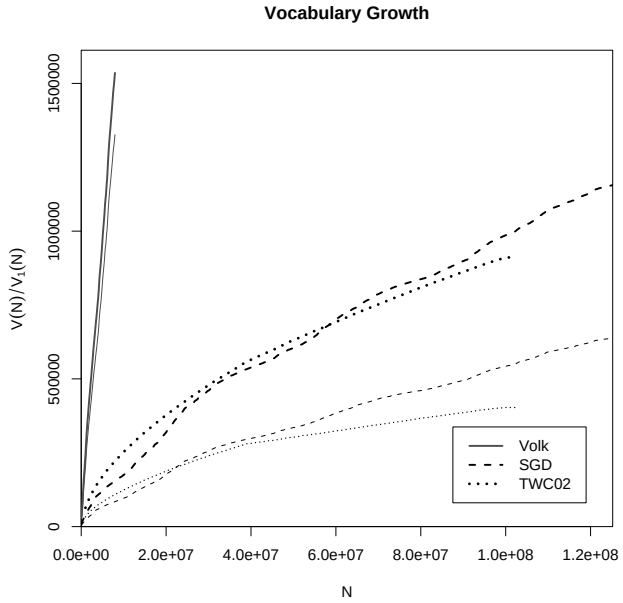


**Fig. 1.** Vocabulary Growth Curves for the contemporary reference corpus Twente Newspaper Corpus 2002 or TWC02, for the contemporary OCRed parliamentary debates corpus or SGD and the historical newspaper corpus 'het Volk, articles, 1918'. Attendant finer lines depict the growth in terms of the hapax legomena.

**Table 1.** Corpora Statistics: Corpus, language (CD: Contemporary Dutch, HD: Historical Dutch), origin: born-digital (BD) or OCRed (OCR), number of word tokens, number of word types, type-token ratio (TTR)

| Corpus | Lang. | Origin | Tokens | Types | TTR |
|---|---|---|---|---|---|
| TWC2 | CD | BD | 92,793,519 | 914,026 | 0.985% |
| SGD | CD | OCR | 125,209,007 | 1,156,998 | 0.924% |
| DDD | HD | OCR | 7,950,950 | 1,535,529 | 19.31% |

highly systematic, are anything but. High systematicity in particular confusions is present, allowing for some variants to accrue high token frequencies. On the whole, however, just about anything may happen, although less frequently. To illustrate this, Table 2 lists twenty variants for 'regeering' which differ only in the last character, the substitutions observed covering nearly the full alphabet.

[4] presented statistics on large collections of typographical errors (English: 12,094 typos and Dutch: 9,152) collected in large corpora. The bulk of the errors there were shown to be single-character insertions, deletions and substitutions, in line with the findings of previous studies, the largest of which was [5]. In Table 3 and Table 4 we list comparable statistics obtained from the OCRed corpora we here work with: statistics on 5,047 mainly OCR-errors from the SGD and 3,799 from the DDD. For the SGD we focused on the word 'belasting' ('tax'),

**Table 2.** Twenty variants (multiple non-contiguous errors) for the focus word 'regeering' produced by apparent random substitutions of the focus word's last character(s), besides the recurring substitution of an 'e' by 'c'

| | | | | | | |
|---|---|---|---|---|---|---|
| regecrin | regecrinc | regecring' | regecrinj | regecrino | regecrins | regecrinz |
| regecrin- | regecrincr | regecrini | regecrink | regecrinp | regecrint | regecrinü |
| regecrina | regecrinf | regecrinic | regecrinn | regecrinr | regecrinx | |

a common topic in parliamentary debate, and strove to identify all variants in all morphological and compound forms of the word. In all, we identified 1,577 variants for the various guises of the noun 'belasting'. For the DDD we opted to identify all the variants for the lemma 'regeering', i.e. 'government'. This lemma yielded 1,468 variants in a single newspaper in the DDD, the 1918 edition of 'Het Volk' alone.

The statistics clearly show the qualitative differences between typewritten, i.e. born-digital, and OCRed text collections. The bulk of the errors found in the latter are substitutions and multiple errors. A multiple error cannot be described by reference to just one of the 4 categories of error, i.e. either to insertion, deletion, transposition or substitution [6], alone. A multiple contiguous error (multi-C) would be the OCR-error 'regeermg' for 'regeering' (6 hits on the DDD website[3], i.e. the multiple error consisting of deletion of an 'i' and substitution of the 'n' by 'm' is situated in one location within the word. A multiple non-contiguous error (multi-NC) would be the OCR-error 'rcgecring' for 'regeering' (243 hits). These statistics further show us what is expected from a system geared at correcting OCR-errors. While the smaller edits required for typewritten text are present, an OCR-oriented correction mechanism has to be able to deal with greater edits, often located in several locations within a particular word string.

## 3 Related Work

### 3.1 Historical Text Collections: Prior Work

Most approaches to historical spelling variation for correction and/or text retrieval attempt to model the historical typographical variation observed by means of rules, either created manually or derived (semi-)automatically ([7] for English, [8], [9], [10] for German and [11] for Dutch). These authors typically work on older, pre-standardization era language variants than we do here. The spelling De Vries-Te Winkel can be seen as one of the first attempts at spelling standardization for Dutch. In this prior work very little attention, if any, is devoted to the typographical variation due to the OCR-process. This is either because the corpus was rekeyed or because the text underwent thorough editing after OCRing. [10] mention that OCR-errors are a problem, but do not address

[3] URL: http://kranten.kb.nl/index4.html (deselect: 'Het Centrum' and 'De NRC')

**Table 3.** SGD 1989-1995: overview and statistics per LD of error-types encountered in a sample of 5,047 non-word variants

| Category | LD 1 | LD 2 | LD 3 | LD 4 | LD 5 | LD 6 | LD 7 | Total | % |
|---|---|---|---|---|---|---|---|---|---|
| deletion | 221 | 10 | 3 | 1 | | | | 235 | 4.66 |
| insertion | 1,980 | 27 | 6 | 11 | | | | 2,024 | **40.10** |
| substition | 1,065 | 49 | 37 | 3 | | 1 | | 1,155 | **22.89** |
| transposition | | 26 | | | | | | 26 | 0.52 |
| multi-C | | 722 | 30 | 10 | 1 | 1 | | 779 | **15.46** |
| multi-NC | | 303 | 271 | 101 | 22 | 5 | 2 | 710 | **14.09** |
| run-on words | 67 | | | | | | | 67 | 1.33 |
| split word | 32 | | | | | | | 32 | 0.63 |
| TOTAL | 3,380 | 1,138 | 347 | 126 | 23 | 7 | 2 | 5,047 | |
| % | 66.98 | 22.55 | 6.88 | 2.50 | 0.46 | 0.14 | 0.04 | | 100.00 |

**Table 4.** DDD 'Het Volk' 1918: overview and statistics per LD of error-types encountered in a sample of 3,799 non-word variants

| Category | LD 1 | LD 2 | LD 3 | LD 4 | LD 5 | LD 6 | Total | % |
|---|---|---|---|---|---|---|---|---|
| deletion | 31 | 27 | 1 | 12 | | | 71 | 1.87 |
| insertion | 133 | 25 | 3 | 4 | | | 165 | 4.34 |
| substition | 575 | 276 | 109 | 2 | | | 962 | **25.32** |
| transposition | | 3 | | | | | 3 | 0.08 |
| multi-C | | 203 | 193 | 9 | 2 | 1 | 412 | **10.85** |
| multi-NC | | 810 | 1,277 | 77 | 15 | 3 | 2,182 | **57.44** |
| run-on words | 2 | | | | | | 2 | 0.05 |
| split word | 2 | | | | | | 2 | 0.05 |
| TOTAL | 743 | 1,344 | 1,583 | 104 | 17 | 4 | 3,799 | |
| % | 19.56 | 35.38 | 41.67 | 2.74 | 0.45 | 0.11 | | 100.0 |

it. The work of [7] discusses adapting Aspell for work with 18th century English. Other interactive systems for post-correction of OCRed texts are described in [12], which was based on Ispell and [13], who describe an elaborate system for training an interactive OCR post-correction tool.

The digitization project underway at the National Library comprises an estimated 8 million pages of newspaper text, good for an estimated 25 billion words of running text. The sheer scale of digitization envisaged there and by similar institutions around the globe precludes even considering interactive post-correction of the OCRed corpora.

### 3.2 Large Dictionary Word Variant Retrieval: Prior Work

[14] propose a Universal Levenshtein Automata based approach to fast approximate search in large dictionaries. We note they evaluate exclusively on average

retrieval times of the sets of correction candidates per length class of focus words. The system is claimed to perform exhaustive variant retrieval up to LD 3, but this is not supported by evaluation. The method is shown to be largely language-independent, but requires training and is sensitive to the size of the dictionary and the number of correction candidates, i.e. variants within the LD limit (the 'bound' in their terms), present. Given that the authors work with randomly introduced errors for all three languages, an assessment of the system's performance on both recall and precision would have been straightforward: if one considers the original list to be the gold standard and one introduces errors, one has a benchmark set against which to gauge one's system. We note the omission with wonder. This is remedied to some extent in [15] where recall scores are in fact given. The system was extended to return only restricted candidate sets. The results suggests that in fact the system does not exhaustively return all the correction candidates. This is particularly evident in the scores on LD 1, which shows the desirability of evaluating one's system not only globally, but also looking more specifically at specific ranges, here sets of variants sharing the same LD to their canonical form. Testing on lists containing only erroneous word forms, they could not perform an evaluation in terms of precision, although this is not mentioned. [16] apply the system to classify English and German webpages according to the level of lexical errors they contain. The paper discusses in full how the error dictionaries that are used for this purpose are built. We note that the system is strictly limited to LD 3. Further that by their own calculations over one third of the actual errors present in webpages is not detected: on lists of 1,000 real-world errors collected from webpages, over 37% for English and over 36% for German are not accounted for by the error dictionaries.

## 4 OCR Post-Correction

### 4.1 Preliminaries

In contrast to the above cited recent work on historical text collections, we here do not focus almost exclusively on the historical variation, but almost exclusively on the variation caused by the OCR-process. This is because the spelling variation due to historical change is almost negligible compared to the OCR-variation in the DDD.

The system we propose, TICCL, is an unsupervised, scalable, fully automatic solution which requires no training (apart from some unavoidable text preprocessing) and which is largely language-independent. In fact, this approach should work for most alphabetical languages. TICCL does not try to account for unknown word types, i.e. words not present in the lexicon, in terms of their likely in-vocabulary counterpart. Rather, it tries to exhaustively gather all the likely non-word variants for a given known or, if unknown, given high frequency word type. This entails the system does not operate on the tokens in running text, but rather on the word type list derived from the corpus to be post-corrected. The system can be run with or without an extra validated word lexicon for the

language. If no validated word lexicon is available, a word type list derived from a background corpus for the language may be used instead. The work by [17] is relevant in this respect because it shows that existing OCR-systems can be used to bootstrap resources for languages for which no customised OCR-systems exist or for which resources are scarce. Historical Dutch text can today be seen as an under-resourced language. For the historical spelling we work with here, we do not have a validated lexicon at our disposal nor was the OCR-system used adapted for older variants of Dutch.

### 4.2 Anagram Hashing

We propose an adaptation of the core correction algorithm we have described in depth in [18]. Anagram Hashing first uses a bad hashing function to identify all word strings in the corpus at hand that consist of the same subset of characters and assigns a large natural number to them, to be used as an index. Informally, the numerical value for a word string is obtained by summing the ISO Latin-1 code value of each character in the string raised to a power $n$, where $n$ is empirically set at: 5. In effect, all anagrams, words consisting of a particular set of characters and present in the list, will be identified through their common numerical value. As the collisions produced by this function identify anagrams, we refer to this as an **anagram hash** and to the numerical values obtained as the **anagram values** (henceforth: AVs) and **anagram keys**, when we discuss these in relation to the hash. Based on a word form's anagram key it thus becomes possible to systematically query the list for any variants present, be they morphological, historical, typographical or orthographical.

The 'alphabet' used when we allow the system to search up to three (and more) character edits, contains the AVs for single characters and all possible two-character and three-character combinations. We call this the AnagramValueAlphabet. Note that a single value in this alphabet represents a single character or any combination of two (2 combinations) or three (6 combinations) particular characters, allowing for efficient look-up. The focus word is not likely to contain all the characters in the AnagramValueAlphabet. The subset of values from the AnagramValueAlphabet derivable from the characters actually present in the focus word forms the FocusWordAlphabet.

The lexicon is a regular hash built up at run-time having the AVs as keys and chained anagrams as values. We use the AV for the focus word and the FocusWordAlphabet and AnagramValueAlphabet to query the lexicon for variants of the focus word. These variants can all be seen as variations and combinations of the usual error type taxonomy. In our implementation, all four edit operations are handled as substitutions. For **substitutions**, a value from the FocusWordAlphabet is subtracted and a value from the AnagramValueAlphabet added. A single query on the AV for 'regeering' minus the AV for an 'e', plus the AV for a 'c' may thus retrieve the three OCR-errors: 'rcgeering', 'regcering' and 'regecring'. **Insertions** are substitutions where a value from the FocusWordAlphabet is subtracted and zero added. **Deletions** are substitutions where zero is subtracted

and a value from the AnagramValueAlphabet added. To find **transposition** errors nothing needs to be added or subtracted.

By systematically querying the lexicon hash we retrieve all possible variants that fall within reach. The actual reach is defined by the alphabet used, i.e. depends on whether the alphabet contains the AVs for single characters only, or also for the full set of character bigrams, or even trigrams. The actual number of hash look-ups required is defined by the number of unique values in the AnagramValueAlphabet and by the number of unique values for all the character combinations in the focus word.

The anagram-based core correction system proposed by us in [18] was here extended on the basis of conclusions we were allowed to draw concerning the nature of OCR-errors on the basis of the statistics gathered from lexical variants culled from the SGD and DDD. In fact, only relatively minor extensions proved necessary. Since our earlier system had been developed to handle typos only, it had no good provisions for handling multiple non-contiguous errors. This shortcoming towards OCR-errors was remedied by extending the focus word derived anagram values to allow for all character n-grams, where n <= 3, to be derived and used in the search for variants, rather than only the contiguous character n-grams. Further, the algorithm's reach in terms of Levenshtein distance was extended by allowing for every transformation to occur twice. This is motivated by the fact that the OCR-process is likely to misrecognize e.g. the character bigram 'in' and to render it as 'm'. If 'in' occurs twice within a particular word, this may then very well happen twice. It is possible to further extend the algorithm towards exhaustive LD 4 and higher retrieval. This necessarily comes at a cost.

The pseudo-code for our adapted and extended algorithm is presented next:

```
Set LDlimit to 3
    Foreach CounterOne in (1,2)
        Foreach CounterTwo in (1,2)
            For each AlphabetValue in Alphabet
                For each FocuswordAlphabetValue in FocuswordAlphabet
                NewValue = FocuswordAnagramValue − (FocuswordAlphabetValue
                × CounterOne) + (AlphabetValue × CounterTwo)

                    If NewValue defined in LexiconHash
                    VARIANTS = list of string variants associated with NewValue
                    Get WordLength of FocusWord
                    If CounterOne = 2 or CounterTwo = 2 and WordLength > 9
                    LDlimit = LDlimit × 2
                    Endif
                    If WordLength < 7
                    LDlimit = 2
                    Endif
                    Else
                    LDlimit = 3
                    Endelse
```

```
                    For each variant in VARIANTS
                    Calculate Levenshtein Distance between focus and variant
                        If LD <= LDlimit
                        Return variant
Endif ; Endfor ; Endif ; Endfor ; Endfor ; Endfor ; Endfor
```

### 4.3 Character Normalization

Preliminary experiments were performed on the SGD. The SGD's higher OCR-accuracy (largely due to better print quality, modern fonts more suited to the OCR-system, clearer page layout) estimated by the National Library at 99% provided a gentler introduction to the peculiarities of post-correction of OCR-errors than the DDD, estimated at 70%, would have. In these initial experiments we worked with the full alphabet encountered within the corpus. In the SGD we observed that 187 characters were actually in use. Given the far larger search space created by the OCR-process in the DDD, we in later stages decided to opt for extensive character normalization. This was performed according to the following scheme: non-visible, non-printing characters were simply discarded. All text is lowercased. All digits and numbers are rewritten as a single '3'. All punctuation marks, except hyphens and apostrophes, are rewritten as a single '2'. Uppercased characters bearing diacritics are rewritten as '4'. Lowercased characters bearing diacritics are rewritten as '5'. This leaves us an alphabet of just 32 characters. Normalization is not a necessary step, but it constricts the search space and promotes the scalability of the system.

Tokenization on low-accuracy OCRed text is not a good idea since all words into which the process inserted punctuation are then split. For performance evaluations for text processing of noisy inputs, please refer to [19]. The frequency lists are in this new scheme obtained by regarding any string delimited by spaces as a word string, instead of after tokenizing.

### 4.4 Processing Steps

The system first reads in the validated lexicon, if available. Words in the validated lexicon are currently invariably rejected as variants during further processing, i.e. we do not do real-word correction. For the tests reported here, this validated lexicon was primarily based on a new open-source spelling checking dictionary[4] containing not only lemmas, but also expanded word forms. TICCL next reads in the word unigram frequency list derived from the corpus to be cleaned. The validated lexicon has no frequency information, but any word type already logged there inherits the frequency information from the corpus-derived list. Next, TICCL reads the list of focus words to be processed, typically a range of frequency/word length values covering a part or the whole of the corpus to be cleaned. If available, TICCL further reads in a list of pre-processed variants (merged and split words, variants containing superfluous hyphens, etc. obtained

[4] URL: http://www.opentaal.org/

by a script that uses word uni- and bigrams to tackle these specific problems) in order to add these to the other variants retrieved for a particular focus word during normal processing. While TICCL reads in the accumulated lexicons to be processed, it applies the character normalization steps, retaining all surface forms with their normalized version. TICCL next gathers all the variants present within the lexicon for the focus words it is set to work on and applies the Levenshtein distance limit filter to discard variants exceeding the LD limit and the validated lexicon filter and text-induced morphological variants filter to not retain real-word variants. TICCL finally returns the retained variants to an output file in the form of pairs: (focus word, retrieved variant).

## 5 Evaluation

### 5.1 Test Sets - How We Evaluate

We evaluate on a subset of the paired lists of variants and focus word for which we presented error distribution statistics in Section 2. The subsets involved all the variants for the 20 SGD focus words in Table 5, 890 in all, and all the variants for the 17 focus words for 'Het Volk', 3,102 in all. Listed next to the focus words are the numbers of variants found. We have had to rely on sampling the typographical variation present within the corpora, exhaustively gathering all the typographical variants for the focus words. As can be seen there is some overlap in the common words between the SGD and 'Het Volk'. Names, especially names of historical figures, being more tied to their era, provide less opportunity for such overlap.

We evaluate in terms of recall and precision, resulting in the combined F-score [20]. These metrics are derived from the numbers of True Positives (TPs), False Positives (FPs) and False Negatives (FNs) returned by the system. **True Positives** are defined by what constitutes the **target** of our exercise. The target for TICCL are the non-word variants (be they typos or OCR-errors) present in the corpus-derived list to be processed. **False Positives** are non-word variants, or real-words absent from the lexicon, that are erroneously reported to be variants for a particular focus word. **False negatives** are those items in the list of known, annotated variants[5] for the particular focus word that are absent from the list of variants returned, i.e. that the system was not able to retrieve or 'correct'.

The scores we present in Table 6 and Table 7 are the scores obtained per LD. The sum of True Positives and False Negatives gives the total amount of variants on which we evaluated. We list the Recall R, Precision P and F-scores F obtained at each LD and complement these with the Cumulative Recall CR, Cumulative Precision CP and Cumulative F-scores CF to the particular LD. The score at a particular LD is thus measured for each variant retrieved which is 1 or 2 or 3 or more edits removed from the focus word. The cumulative score is based on the accumulated numbers of TPs, FNs and FPs observed at each LD up to the LD under scrutiny. The formulae used are as follows:

[5] This motivates why we only evaluate on subsets of our collections of annotated variants: all variants for a particular focus word need to be annotated, exhaustively.

**Table 5.** Overview of the SGD and DDD focus words and their observed numbers of variants which constitute the evaluation sets. Capitalized words are proper names

| Focus SGD | # | Focus 'Het Volk' | # |
|---|---|---|---|
| Achtienribbe-Buijs | 23 | Amsterdam | 307 |
| Amsterdam | 43 | Annexionisten | 20 |
| Bolkestein | 18 | België (Belgium) | 104 |
| Jorritsma-Lebbink | 33 | Bismarck | 10 |
| Nieuwenhoven | 22 | Compiègne | 3 |
| Rotterdam | 47 | Hindenburg | 32 |
| Wolffensperger | 25 | Nederlandsche (Dutch) | 572 |
| belasting (tax) | 36 | Posthuma | 264 |
| belastingen (taxes) | 56 | Richthofen | 7 |
| belastingplichtige (taxable person) | 41 | Trotzky | 45 |
| belastingplichtigen (taxable persons) | 37 | Wilhelmina | 42 |
| doelstelling (aim) | 82 | Zeeuwsch-Vlaanderen | 19 |
| doelstellingen (aims) | 58 | belasting (tax) | 102 |
| evaluatie (evaluation) | 44 | belastingen (taxes) | 34 |
| faciliteiten (facilities) | 27 | distribueeren (to distribute) | 52 |
| goedkeuring (approval) | 36 | eenheidsworst (unity sausage) | 21 |
| inkomstenbelasting (income tax) | 81 | regeering (government) | 1468 |
| motorrijtuigenbelasting (motor vehicle tax) | 70 | | |
| studiefinanciering (study financing) | 93 | | |
| vennootschapsbelasting (corporate tax) | 52 | | |

$$\text{Recall} = \text{R} = \frac{TP}{TP+FN} \qquad \text{Precision} = \text{P} = \frac{TP}{TP+FP}$$

Since we deem recall and precision to be equally important, the harmonic mean of R and P, the simplified F measure, F, is given by:

$$\text{F-score} = \text{F} = \frac{2 \times R \times P}{R+P}$$

**Recall** expresses to what extent TICCL has been able to identify the non-words (typos and OCR-errors) in the full corpus derived list. **Precision** then expresses to what extent TICCL has been able to assign an identified variant to its proper canonical form. In other words, we desire the list of possible variants returned by TICCL to contain as many as possible of the actual variants present in the corpus-derived word type list for a particular focus word and as few as possible of the variants actually belonging to another focus word. Note that in these evaluations we are very strict: a variant reported for e.g. the singular form 'tax' which in fact in the original text read 'taxes', will be counted as an FP. Note too that precision scores provide information about the coverage of the lexicon used. In the SGD we find the hapax legomenon 'gelasting' which should be understood as 'court order', i.e. a domain specific kind or 'order'. The word is absent from the validated lexicon we used, is returned as a variant for 'belasting

('tax') and consequently counted as an FP. The Zipfian nature of the distribution of errors over the LDs entails that the cumulative scores at higher LDs remain relatively high: there being fewer errors at higher LDs means that the relative proportion of these in the cumulative score is smaller, i.e. they have less impact on the cumulative score. It should be noted that our scores do not take into account the token frequencies of the variants retrieved. The scores reported here are scores on word types only, so each retrieved variant has an equal share in the scores obtained. Weighting the scores by token rate would in fact likely make the scores look better, in so far that [21] predicts that small LD accidents happen far more often than larger LD accidents.

**Table 6.** Overview of the SGD performance scores

| Measured at | Items retrieved | | | At LD | | | Cumul. to LD | | |
|---|---|---|---|---|---|---|---|---|---|
| LD | TP | FN | FP | R | P | F | CR | CP | CF |
| LD 1 | 466 | 4 | 7 | 0.991 | 0.985 | 0.988 | 0.991 | 0.985 | 0.988 |
| LD 2 | 284 | | 129 | 1.000 | 0.688 | 0.815 | 0.995 | 0.847 | **0.915** |
| LD 3 | 106 | 1 | 525 | 0.991 | 0.168 | 0.287 | 0.994 | 0.564 | 0.720 |
| LD 4 | 11 | 11 | 133 | 0.500 | 0.076 | 0.133 | 0.982 | 0.522 | 0.682 |
| LD 5 | 1 | 6 | 22 | 0.143 | 0.043 | 0.067 | 0.975 | 0.515 | 0.674 |

**Table 7.** Overview of the DDD performance scores

| Measured at | Items retrieved | | | At LD | | | Cumul. to LD | | |
|---|---|---|---|---|---|---|---|---|---|
| LD | TP | FN | FP | R | P | F | CR | CP | CF |
| LD 1 | 380 | 6 | 4 | 0.984 | 0.990 | 0.987 | 0.984 | 0.990 | 0.987 |
| LD 2 | 1112 | 9 | 114 | 0.992 | 0.907 | 0.948 | 0.990 | 0.927 | **0.957** |
| LD 3 | 1558 | 3 | 613 | 0.998 | 0.718 | 0.835 | 0.994 | 0.807 | 0.891 |
| LD 4 | 25 | 9 | 46 | 0.735 | 0.352 | 0.476 | 0.991 | 0.798 | 0.884 |

### 5.2 Discussion Of Evaluation Results - Future Work

In collecting the evaluation sets we did not strive at obtaining a full balance between the proportion of names included (SGD: 35%, DDD: 70.6%). This discrepancy explains why our system apparently performs better on lesser quality OCRed text than on far better quality OCRed text. In fact, performance on at least some of the names is markedly better than on common words. For some names, there simply are no other words resembling them to the extent that these would fall within the LD of 1, 2 and even more edits. As such, words with a higher neighbourhood density [22], especially short words and words derived from a stem and highly common pre- and/or affixes such as e.g. 'belasting', are far more likely to incur more False Positives.

We have here reported solely on TICCL's capabilities of retrieving a focus word's variants and have reported these in terms of both Recall and Precision. We believe TICCL in this guise constitutes a usable and useful system: if it is set to work within the limits of LD 2, given our statistics on the actual distribution of OCR-errors within these text collections, our scores in effect mean that for the SGD almost 89% and for 'Het Volk' almost 55% of the less desirable typographical variation in terms of non-words present can now fully automatically be removed, as these are the summed percentages of Levenshtein distance 1 and 2 errors observed in our 5,047 SGD and 3,799 'Het Volk' error samples. These results are obtained by using only very simple but efficient and effective retrieval and filtering strategies. In a future paper we hope to report on how by extending our system with substring processing capabilities (Dutch compounds being written as single words) and by taking account of context, we achieve higher precision scores on higher LD errors.

## 6 Conclusion

We believe to have demonstrated that we have a built a competitive system which can greatly enhance the overall quality of large corpora of OCRed text aimed at text retrieval. The system being inherently simple, sufficiently efficient to scale to very large corpora and yet not language-dependent in se, we hope to see it adopted in large scale digitization programmes around the world and adapted to local needs. To further this, we intend to release TICCL to the Open Source community in due course.

*Acknowledgments* This work was funded by the Netherlands Organisation for Scientific Research (NWO).

## References

1. Baayen, R.H.: The effects of lexical specialization on the growth curve of the vocabulary. Computational Linguistics **22** (1996) 455–480
2. Evert, S., Baroni, M.: zipfR: Word frequency distributions in R. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Posters and Demonstrations Session, Prague, Czech Republic (2007)
3. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, Massachusetts; London, England (1999)
4. Reynaert, M.: Corpus-Induced Corpus Clean-up. In: LREC 2006: Fifth International Conference on Language Resources and Evaluation, Magazzini del Cotone Conference Center – Genova, Italy, Paris : ELRA, European Language Resources Association (2006)
5. Pollock, J., Zamora, A.: Collection and characterization of spelling errors in scientific and scholarly text. Journal of the American Society for Information Science **34** (1983) 51–58
6. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Communications of the ACM **7** (1964) 171–176

7. Schneider, P.: Computer assisted spelling normalization of 18th century English. Language and Computers **36** (20 November 2001) 199–211(13)
8. Ernst-Gerlach, A., Fuhr, N.: Retrieval in text collections with historic spelling using linguistic and spelling variants. In: JCDL '07: Proceedings of the 2007 conference on Digital libraries, New York, NY, USA, ACM Press (2007) 333–341
9. Pilz, T., Luther, W., Fuhr, N., Ammon, U.: Rule-based search in text databases with nonstandard orthography. Literary and Linguistic Computing **21** (2006) 179–186
10. Hauser, A., Heller, M., Leiss, E., Schulz, K.U., Wanzeck, C.: Information access to historical documents from the Early New High German Period. In Burnard, L., Dobreva, M., Fuhr, N., Lüdeling, A., eds.: Digital Historical Corpora - Architecture, Annotation, and Retrieval. Dagstuhl Seminar Proceedings, Dagstuhl, Germany, IBFI (2007)
11. Adriaans, F., Koolen, M., Kamps, J., de Rijke, M.: A cross-language approach to historic document retrieval. In Lalmas, M., MacFarlane, A., Rüger, S., Tombros, A., Tsikrika, T., Yavlinsky, A., eds.: Advances in Information Retrieval: Proceedings 28th European Conference on IR Research (ECIR 2006). Volume 3936 of LNCS., Springer (2006) 407–419
12. Taghva, K., Stofsky, E.: OCRSpell: an interactive spelling correction system for OCR errors in text. International Journal on Document Analysis and Recognition **3** (2001) 125–137
13. Strohmaier, C.M., Ringlstetter, C., Schulz, K.U., Mihov, S. In: A visual and interactive tool for optimizing lexical postcorrection of OCR-results. IEEE Computer Society, Los Alamitos, CA, USA (2003)
14. Mihov, S., Schulz, K.U.: Fast approximate search in large dictionaries. Journal of Computational Linguistics **30** (2004) 451–477
15. Mihov, S., Mitankin, P., Gotscharek, A., Reffle, U., Schulz, K.U., Ringlstetter, C.: Tuning the selection of correction candidates for garbled tokens using error dictionaries. In: Finite State Techniques and Approximate Search, Proceedings of the First Workshop on Finite-State Techniques and Approximate Search, Borovets, Bulgaria (2007) 25–30
16. Ringlstetter, C., Schulz, K.U., Mihov, S.: Orthographic errors in web pages: Toward cleaner web corpora. Computational Linguistics **32** (2006) 295–340
17. Kolak, O., Resnik, P.: OCR post-processing for low density languages. In: HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Morristown, NJ, USA, Association for Computational Linguistics (2005) 867–874
18. Reynaert, M.: Text-Induced Spelling Correction. PhD thesis, Tilburg University (2005)
19. Lopresti, D.: Performance evaluation for text processing of noisy inputs. In: SAC '05: Proceedings of the 2005 ACM symposium on Applied Computing, New York, NY, USA, ACM Press (2005) 759–763
20. van Rijsbergen, C.J.: Information Retrieval. Butterworths, London (1975)
21. Zipf, G.K.: The psycho-biology of language: an introduction to dynamic philology. 2 edn. The M.I.T. Press, Cambridge, MA (1935)
22. Frauenfelder, U., Baayen, R., Hellwig, F., Schreuder, R.: Neighbourhood density and frequency across languages and modalities. Journal of Memory and Language **32** (1993) 781–804

This article was processed using the LaTeX macro package with LLNCS style