Poor Man's OCR Post-Correction: Unsupervised Recognition of Variant Spelling Applied to a Multilingual Document Collection

Poor Man's OCR Post-Correction

Anonymous Author(s)
Anonymous Affiliation Row 1
Anonymous Affiliation Row 2
Anonymous Country
some@email.address.com

ABSTRACT

The accuracy of Optical Character Recognition (OCR) is sets the limit for the success of subsequent applications used in text analyzing pipeline. Recent models of OCR post-processing significantly improve the quality of OCR-generated text but require engineering work or resources such as human-labeled data or a dictionary to perform with such accuracy on novel datasets. In the present paper we introduce a technique for OCR post-processing that runs off-the-shelf with no resources or parameter tuning required. In essence, words which are similar in form that are also distributionally more similar than expected at random are deemed OCR-variants. As such it can be applied to any language or genre (as long as the orthography segments the language at the word-level). The algorithm is illustrated and evaluated using a multilingual document collection and a benchmark English dataset.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; H.3.6 [**Information Search and Retrieval**]: Library Automation - Large text archives; I.2.7 [**Computing Methodologies**]: [Artificial Intelligence - Natural language processing]; I.5.4 [**Pattern Recognition**]: [Applications - Text processing]

General Terms

OCR, Multilingual

Keywords

OCR, Multilingual, Unsupervised

1. INTRODUCTION

In search engines and digital libraries, more and more documents become available from scanning of legacy print collections rendered searchable through optical character recognition (OCR). Access to these texts is often not satisfactory due to the mediocre performance of OCR on historical texts and imperfectly scanned or physically deteriorated originals [19, 21]. Historical texts also often contain more spelling variation which turns out to be a related obstacle.

Given this need, techniques for automatically improving OCR output have been around for decades. Most, if not all, of these techniques require resources in the form of a dictionary, a morphological analyzer, annotated training data, (re-)tailoring to a specific language, or the fine-tuning of thresholds or other parameters. However, for many practical applications, especially when it comes to multilingual and/or genre-specific text collections, such resources are expensive to produce or cannot be obtained at all. For these applications, a slim, language-independent, off-the-shelf solution is more attractive. This is the motivatation for the present approach.

Given only raw-text data, distributional similarity (e.g., via Word2Vec) as well as form similarity (e.g., edit distance) between types can be computed efficiently. The present paper presents an OCR post-correction algorithm based simply on juxtaposing these two measures: two words are variants of each other if their distributional similarity exceeds that expected by chance from their form similairity. The algorithm uses no language-specific information, runs in quadratic time and requires no thresholds or parameters (unless such are used to obtain the form and/or distributional similarity).

The present paper first summarizes the state-of-the-art in OCR post-correction in Section 2. The poor man's OCR correction algorithm is described and exemplified in Section 3. Experiments and evaluation are detailed in Section 4, followed by a discussion of strengths, weaknesses and possible enhancements of the present approach in Section 5.

2. RELATED WORK

Algorithms applicable to the OCR post-correction problem were developed as early as [5], though the target was the similar problem of spelling correction rather than OCR correction per se. A concise survey of trends in subsequent work can be found in [14, 6-13] [15, 1347-1348] and need not be repeated here; instead we summarize the current state-

of-the-art. OCR post-correction systems suggest corrections based on form similarity to a more frequent word, and, if a dictionary of correct forms is available, positive use can be made of it [7]. Pairs of words with a given edit distance can be found efficiently, in sub-quadratic running time [17, 4]. A number of features based on form, frequency and dictionary properties may be used to rank candidates [8]. Most systems use labeled training data and treat the task of ranking candidates as a standard supervised Machine Learning problem [15, 12] though [1, 2] treat it as an SMT problem and thereby make some use of context. A few systems use context explicitly [20, 8] and the work in the present paper can be said to continue this line. However, none the systems so far described can be run off-the-shelf; some resources or human interaction is required to produce corrected OCR output. The systems covered in [1, 2, 15, 12, 7] necessitate language-specific labeled training data, while [9, 10] need a dictionary, and [3] relies on google to provide a dictionary. The approaches by [8, 18, 20] need some human intervention to set dataset-specific thresholds or some additional component to choose among alternatives. [11] requires several independent OCR engines and alignment of their output.

Unfortunately, there appears no be no widely used gold standard dataset available, so the various systems cannot be straighforwardly compared in terms of accuracy. Given the diversity in prerequisites for the various systems such a comparison with not be conclusive, but it would at least provide some benchmark figures and a starting for cross-system error analysis. One suitable dataset (used in the present paper) is the OCRed newspaper text in English from the Sydney Morning Herald 1842-1954[1] with a manually corrected subset prepared by [8].

3. POOR MAN'S OCR POST-CORRECTION

We start from raw text data as input. We assume that the raw text data can be unproblematically tokenized into words. For the experiments and examples reported here we have tokenized on whitespace recognizing (unicode-)alphanumeric sequences as words.

1. From raw text data we can derive a measure of *distributional similarity*, here denoted $Sim(x, y)$ between terms x, y using a small size window in (now standard) techniques such as Latent Semantic Indexing (LSI) [6] or Word2Vec [13]. The OCR-correction strategy is oblivious to the specific choice of measure for distributional similarity[2].

2. We define a form-neighbourhood function $N(x)$ that gives the set of forms close to x. The exact choice of neighbourhood function depends on ambition, but the natural choice for is the set of forms with edit distance (ED) ≤ 1 to x: $N(x) = \{y | ED(x, y) \leq 1\}$.

3. Now we may define the key criterion $V(x, y)$ which assesses whether y is an OCR variant (or, more generally, spelling/string variant) of x. Fixing on a specific x let $Sim_x(y) = S(x, y)$ be the list of similarity values x has to all other terms and let $Sim_x(y)[k]$ denote the kth highest value. $V(x, y)$ is rendered true iff $S(x, y)$ exceeds that expected by chance from $|N(x)|$ trials from $Sim_x(y)$. In other words, suppose we selected $|N(x)|$ values randomly from $Sim_x(y)$ (the similarity that x has to all other terms), if the actual similarity $S(x, y)$ of y to x exceeds all of those, then (and only then) y is deemed an OCR variant of x. The expectation when choosing k items from a list of n total items is that the maximum of the k values is at the $k + 1$th quantile. For our purposes then, we need to check if $S(x, y)$ is in the $|N(x)| + 1$th quantile of $Sim_x(y)$. We thus define $V(x, y) = S(x, y) > Sim_x(y)[\frac{1}{|N(x)|+1} \cdot |Sim_x(y)|]$.

4. Finding all variants for all terms in the input can now simply be done by computing $V(x, y)$ for each x and its form associates $y \in N(x)$. Each outcome equivalence class of variants may then be normalized ("corrected") to its most frequent term.

5. The $V(x, y)$ answers would be sufficient if it were not for a complication: the existence of high-frequency[3] minimal pairs that are somewhat distributionally similar. Naturally, a language may have forms that are minimal pairs, and it may be that both members in a pair are frequent and some such pairs happen to achieve significant distributional similarity. For example, in English, 'in' and 'on' is a minimal pair with both members frequent and distributionally similar. The strategy described so far would identify the less frequent of the two as an OCR variant of the more frequent one, which is a false positive with dire consequences on the token level. Most OCR post-correction systems use a dictionary or frequency threshold that would whitelist such cases unproblematically. However, in our strive not to rely on such resources, the following strategy allows us to distinguish minimal pairs form OCR-errors heuristically. If a form y is really an OCR error of x, we expect the frequency of y (denoted $f(y)$ to derive from the frequency of its patron x by an error rate r. The appropriate error rate r cannot be known beforehand as it depends on dataset specifics such as the quality of scanned originals. But the OCR error identifications of $V(x, y)$ allow us to at least estimate r_V an upper bound on r for the specific dataset at hand. If x, y are the set of pairs for which $V(x, y)$ is true, the estimate may be formulated as $r_V = \frac{\sum f(y)}{\sum f(x) + \sum f(y)}$. $V(x, y)$ contains the true OCR errors along with the minimal pairs so it constitutes an overestimate of the error rate r. Now, if an individual pair in $V(x, y)$ manifests an even higher error rate than this, it is probably safer to regard them as a minimal pair. If we additionally scale the estimate of the error rate of a potential minimal pair by the distributional similarity (pushing distributionally unsimilar pairs not to be regarded as OCR errors) we arrive at the following formula for filtering $V(x, y)$: $O(x, y) = \frac{f(y)}{f(x)+f(y)} / S(x, y) < r_V$.

The procedure may be iterated, whereby variants identified in the OCR post-correction are conflated and re-fed to

[1] Available at `http://overproof.projectcomputing.com/datasets/` accessed 1 Jan 2017.

[2] We have used the efficient and easily-accessible implementations of LSI and Word2Vec in [16].

[3] Minimal pairs where one or both members are of low-frequency are unlikely to occur and do little harm when they do.

Table 1: Terms for which $O(x, y)$ is true where is x is the term 'language', i.e., terms deemed OCR variants of 'language'.

y	$Sim(x, y)$	$f(y)$	$\frac{f(y)}{f(x)+f(y)}$	$\frac{f(y)}{f(x)+f(y)}/S(x, y)$
ianguage	0.52356	387	0.00066	0.00125
languagc	0.50100	225	0.00038	0.00076
languaqe	0.44455	93	0.00016	0.00035
languuge	0.29799	68	0.00012	0.00038
languago	0.37767	135	0.00023	0.00060
lauguage	0.34320	77	0.00013	0.00038
lunguage	0.46430	63	0.00011	0.00023

the distributional similarity calculation, which in turn may suggest new OCR variants, until convergence (cf. a similar set-up in [15, 1351], where convergence is said to be reached after only a handful of iterations). Especially if the neighbourhood function is conservatively bound to edit distance 1, iteration is the only way to achieve OCR post-correction involving more than one character per term.

We will now illustrate the procedure with an example from the dataset (see below for details) used for experiments .

1. We run Word2Vec with the default settings[4] to get a vector space representation for each term. As an example, the ten terms most similar to 'language' is shown below.

Rank	y	$S(language, y)$
1	languages	0.7619
2	linguistic	0.7555
3	dialect	0.7381
4	community	0.7074
5	history	0.7036
6	culture	0.6995
7	society	0.6704
8	population	0.6636
9	lexicon	0.6542
10	literature	0.6482

2. As a neighbourhood function, we choose all the one-character substitutions with letters from the English lowercase alphabet (Σ). Consider then the term 'language', $N(language) = \{aanguage, banguage, \ldots, language, lbnguage, \ldots\}$ contains $|language| \cdot \Sigma = 8 \cdot 26 = 208$ forms.

3. Which of these 208 forms have a higher than expected distributional similarity $S(language, y)$ to 'language'? The total vocabulary size is 204 002 and on $|N(language)| = 208$ trials the expected quantile to beat is the $\frac{1}{209} \cdot 204002 \approx 976$th quantile. The 976th highest value of $Sim_{language}(y)$, i.e., $Sim_{language}(y)[976]$ is 0.3839. 7 of the members of $N(language)$ (apart from the term 'language' itself) have a distributional similarity to 'language' higher than this (Table 1).

4. Are any of the terms in Table 1 minimal pairs with 'language'? The the upper bound estimate of the error rate for all pairs in $V(x, y)$ in this test set is $r_V = \frac{458626300}{2714497206} \approx 0.16895$. Significant for the terms in question is that the frequency $f(language)$ is very high, at 581 815, yielding much lower error rates, so none of them is judged a minimal pair. Thus, these are deemed OCR errors to be corrected to 'language'. As a comparison, we also show the corresponding calculation for the term 'they' in Table 2. In this case, two of the potential OCR-variants 'then' and 'them' have such high frequencies that it is unlikely that their frequencies are derived from 'they' with a realistic error rate, and so the calculations reject them as being OCR-variants.

[4]For the record, the default settings are CBOW with mean training method, vector dimensionality 100, learning rate 0.025, word window size 5, minimal frequency 5, threshold for random downsampling of higher-frequency words 1e-3, number of noise words word for negative sampling 5, number of epochs 5, batch size 10000.

Table 2: Terms for which $O(x, y)$ is true vs not true (boldfaced) where is x is the term 'they'. The non-boldfaced terms are deemed OCR variants of 'they'.

y	$Sim(x, y)$	$f(y)$	$\frac{f(y)}{f(x)+f(y)}$	$\frac{f(y)}{f(x)+f(y)}/S(x, y)$
then	0.51802	411360	0.25513	**0.49250**
them	0.70800	378516	0.23964	**0.33847**
thcy	0.32272	1256	0.00104	0.00323
thoy	0.42350	1760	0.00146	0.00345
thej	0.29713	292	0.00024	0.00081
theg	0.33179	143	0.00012	0.00035
ihey	0.42526	882	0.00073	0.00172
thev	0.43283	822	0.00068	0.00158
tney	0.29813	174	0.00014	0.00048
tbey	0.39003	1210	0.00101	0.00258
fhey	0.35574	129	0.00011	0.00030

4. EXPERIMENTS

The basis for the experiments is a collection of over 9 000 raw text grammatical descriptions digitally available for computational processing. The collection consists of (1) out-of-copyright texts digitized by national libraries, archives, scientific societies and other similar entities, and (2) texts posted online with a license to use for research usually by university libraries and non-profit organizations (notably the Summer Institute of Linguistics). The collection is thus of the specific genre that whereby one language (the target-language) is given a human-readable grammatical description in another language (the meta-language). For each document, we know the meta-language it is written in (usually English, French, German, Spanish or Mandarin Chinese), the target-language(s) described in it (one of the thousands of minority languages throught the world) and the type of description (comparative study, description of a specific features, phonological description, grammar sketch, full grammar etc). The collection can be enumerated using the bibliographical- and metadata is contained in the open-access bibliography of descriptive language data at `glottolog.org`. The collection spans as many as 96 meta-languages and 4 005 target-languages and we intend use it for automated data-harvesting/profiling of the 4 000 target-languages. The entire collection has been OCRed using different, not individually recorded, OCR engines (mostly various versions of ABBYY Finereader usually set to recognize based on the meta-language) and contains large amounts of OCR errors due to the varying quality and age of the originals.

For experiments on OCR post-correction, we have used the documents written in a (meta-)language with more than

Table 3: Sizes of datasets used in the experiments.

Meta-language		# Doc:s	# Types	# Tokens
English	eng	23 708	23 114 708	380 467 360
French	fra	3 452	3 585 529	86 699 512
German	deu	2 753	2 830 285	38 643 792
Spanish	spa	2 484	2 490 063	84 925 065
Portuguese	por	1 076	716078	14 420 655
Russian	rus	677	293909	50 387 961
Dutch	nld	528	397564	5 849 144
Italian	ita	329	555043	6 058 028
Indonesian	ind	206	166524	2 163 114

Table 4: Type reduction after OCR post-correction

		# Types		
Meta-language		Before	After	Reduction
English	eng	23 114 708	21 681 596	6.2%
French	fra	3 585 529	3 399 081	5.2%
German	deu	2 830 285	2 742 546	3.1%
Spanish	spa	2 490 063	2 373 030	4.7%
Portuguese	por	716 078	707 583	1.2%
Russian	rus	293 909	539	0.2%
Dutch	nld	397 564	394 171	0.9%
Italian	ita	555 043	550 359	0.8%
Indonesian	ind	166 524	164 524	1.2%

100 documents worth of data. The sizes of the subparts of the collection corresponding to these meta-languages are given in Table 3. Mandarin Chinese is excluded as the script does not natively render word-boundaries, which is a prerequisite for the techniques described in this paper.

For maximal simplicity, we apply the poor man's OCR correction algorithm with one-character substitution ($N(x) = \{y|ED(x,y) \leq 1\}$) over one iteration. As we do not have gold standard data on the languages involved (for English, see below), we gauge the effectiveness simply by observing the proportion of terms corrected, as shown in Table 4. The resulting number of OCR corrections are proportional to vocabulary size but do not otherwise show great variation across languages, suggesting that the method is indeed independent of language. The anomalously low result for Russian was checked and is largely due to what must be erroneous character settings for the OCR of a fair share of documents, resulting in roman-script junk rather than actual Cyrillic OCR errors. The datasets have other differences than language per se, i.e., bias towards a certain era, quality of originals and OCR engine performance, so a more detailed study of cross-linguistic differences in OCR correction performance is not straightforward.

As a rigourous evaluation of accuracy and as a benchmark to compare with other methods, the poor man's OCR correction algorithm was applied to the freely available gold standard test set used by [8]. It consists of OCRed newspaper text in English from the Sydney Morning Herald 1842-1954[5] (called dataset 1) with random sampled manually corrected subset (called dataset 2). The full collection of texts consists of 928 170 types / 10 498 979 tokens which we use for "training", i.e., to calculate $O(x,y)$ for all types. The gold standard subset consists of 11650 types / 38226 tokens.

[5]Available at `http://overproof.projectcomputing.com/datasets/` accessed 1 Jan 2017.

Table 5: Evaluation of poor man's OCR post-correction on the Sydney Morning Herald 1842-1954 dataset 2.

	Dataset 2		After OCR correction		
	Types	Tokens		Types	Tokens
	11 650	38 226		11 650	38 226
Correct forms	7 655	32 714	untouched	7 383	32 225
			hypercorr.	272	489
Erroneous forms	3 995	5 512	untouched	3 152	4 276
			corrected	540	874
			adjusted	303	362

Again, for maximal transparency we apply one-character OCR post-correction over one iteration. The results are shown in Table 5, yielding a modest improvement in Word Error Rate from 85.5% to 86.5%. This is significantly lower than the 93.7% achieved by [8]. However, the latter system has a large number of thresholds and requires resources, such as google n-grams, beyond the training set itself. The present system runs off the shelf with no tuning or other intervention required. Furthermore, the bulk of the hypercorrections are in fact morphological variants (the majority being participle versus past tense in verbs forms) which are not harmful in many information retrieval applications. More important are the number of OCR-errors not corrected. Here, a fair share of the erroneous types (2103/3995) are more than one character away from their correct form which by definition cannot be corrected in present approach in one iteration.

5. DISCUSSION

The running time of the poor man's OCR post-correction algorithm is bounded by the number of types quadratically, times the size of the form neighbourhood. Given a maximal word length, the algorithm is thus quadratic in the size of the vocabulary. The quadratic term comes from computing the $|N(x)|+1$th quantile of $Sim_x(y)$ for each x in the vocabulary. The latter step can be done with time proportional to $|N(x)|+1$ (again, a constant given a maximal word length) times the size of the vocabulary, with an efficient algorithm that avoids complete sorting. If a speed-up is needed, the $|N(x)|+1$th quantile can be computed by an approximation through sampling or a randomized algorithm, reducing the complexity of this step to constant or logarithmic. The above analysis concerns the OCR correction algorithm only, which presupposes a distributional similarity measure. If a non quadratic-time algorithm is used there, the entire approach is bounded by this complexity.

The accuracy of the poor man's OCR post-correction algorithm is lower than extant approaches which make use of resources or supervision so its value lies in being language-independent, unsupervised and intervention free. However, there is a class of errors where the present approach outperforms general lexicon-based systems: genre specific terms. For example, the present collection contains abbrevations that are specific to the genre of grammatical descriptions, such as SOV (to denote that a language described has Subject-Object-Verb constituent order), with a common OCR variant SOY, correctly identified as such (in all meta-languages independently) in the present system.

With a one-character neighbourhood and only one itera-

tion the poor man's OCR correction method is rather conservative, not resulting in a large number of false positives. As mentioned, the majority of such cases are in fact morphological variants whose normalization is not harmful for many bag-of-words based tasks in information retrieval. Perhaps a more serious drawback is that the system requires word-boundaries and, in the present formulation, cannot correct OCR-errors that disrupt word boundaries and cannot be applied to languages whose writing systems do not encode word boundaries.

The system as described is oblivious to glyphs, i.e., it does not use any information related to the shape of a character. It is rather obvious that OCR confusion is character sensitive and such information is often explicitly used in OCR systems, e.g., [8, 48-49]. Including such a component in the present system would be straightforward enhancement.

Like many other systems, the present algorithm corrects types, i.e., strings in abstracto, rather than specific occurrences. Here also, there is room for improvement by considering the specific context in which a string occurs. The point is that a correction ay be suppressed dependending on the context, rather than to suggest different alternatives for correction (something which is still a global property).

6. CONCLUSIONS

Searching or extractig information from digitised text can be fundamentally limited by the quality of OCR text. We described a language and genre-independent unsupervised OCR post-correction system which requires no resources or human tuning. The system consistently improves OCR text but has a lower performance than systems which benefit from large training resources. The implementation (less than 50 lines of Python code) is available from `https://github.com/shafqatvirk/contentProfiling`.

7. ACKNOWLEDGMENTS

Anonymous author was supported by unnamed grant.

8. REFERENCES

[1] H. Afli, Z. Qiu, A. Way, and P. Sheridan. Using SMT for OCR error correction of historical texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*, pages 962–966, 2016.

[2] H. Afli and A. Way. Integrating optical character recognition and machine translation of historical documents. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 109–116, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[3] Y. Bassil and M. Alwani. OCR post-processing error correction algorithm using google online spelling suggestion. *Journal of Emerging Trends in Computing and Information Sciences*, 3(1):90–99, 2012.

[4] L. Boytsov. Indexing methods for approximate dictionary searching: Comparative analysis. *Journal of Experimental Algorithmics*, 16(1):1–91, May 2011.

[5] F. J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, 1964.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[7] S. Eger, T. vor der Brück, and A. Mehler. Statistical learning for OCR text correction. *The Prague Bulletin of Mathematical Linguistics*, 105:77–99, 2016.

[8] J. Evershed and K. Fitch. Correcting noisy ocr: Context beats confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, pages 45–51, New York, NY, USA, 2014. ACM.

[9] K. Kettunen. Keep, change or delete? setting up a low resource ocr post-correction framework for a digitized old finnish newspaper collection. In D. Calvanese, D. De Nart, and C. Tasso, editors, *Digital Libraries on the Move: 11th Italian Research Conference on Digital Libraries, IRCDL 2015, Bolzano, Italy, January 29-30, 2015, Revised Selected Papers*, pages 95–103. Springer International Publishing, Cham, 2016.

[10] I. Kissos and N. Dershowitz. Ocr error correction using character correction and feature-based word classification. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 198–203, Los Alamitos, CA, USA, 2016. IEEE Computer Society.

[11] W. B. Lund, E. K. Ringger, and D. D. Walker. How well does multiple ocr error correction generalize? In *Proceedings og SPIE 9021, Document Recognition and Retrieval XXI*, volume 9021, pages 1–13, 2013.

[12] J. Mei, A. Islam, Y. Wu, A. Moh'd, and E. E. Milios. Statistical learning for OCR text correction. *CoRR*, abs/1611.06950, 2016.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[14] K. Niklas. Unsupervised post-correction of ocr errors. Master's thesis, Leibniz Universitĺat Hannover, 2010.

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

[16] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[17] M. Reynaert. Synergy of nederlab and @philostei: Diachronic and multilingual text-induced corpus clean-up. In N. C. C. Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1224–1230, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).

[18] M. Reynaert. OCR post-correction evaluation of early

dutch books online - revisited. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*, pages 967–974, 2016.

[19] K. Taghva, T. Nartker, and J. Borsack. Information access in the presence of ocr errors. In *Proceedings of the 1st ACM Workshop on Hardcopy Document Processing*, HDP '04, pages 1–8, New York, NY, USA, 2004. ACM.

[20] X. Tong and D. A. Evans. A statistical approach to automatic ocr error correction in context. In *Fourth Workshop on Very Large Corpora*, pages 88–100, 1996.

[21] M. C. Traub, J. van Ossenbruggen, and L. Hardman. Impact analysis of OCR quality on research tasks in digital archives. In *Research and Advanced Technology for Digital Libraries - 19th International Conference on Theory and Practice of Digital Libraries, TPDL 2015, Poznań, Poland, September 14-18, 2015. Proceedings*, pages 252–263, 2015.