Preprint of article accepted to the ECDL 2009.

Improving OCR Accuracy for Classical Critical Editions

Federico Boschetti, Matteo Romanello, Alison Babeu, David Bamman, and Gregory Crane

Tufts University, Perseus Digital Library, Eaton 124, Medford MA, 02155, USA

Abstract. This paper describes a work-flow designed to populate a digital library of ancient Greek critical editions with highly accurate OCR scanned text. While the most recently available OCR engines are now able after suitable training to deal with the polytonic Greek fonts used in 19th and 20th century editions, further improvements can also be achieved with postprocessing. In particular, the progressive multiple alignment method applied to different OCR outputs based on the same images is discussed in this paper.

1 Introduction

The new generation of Greek and Latin corpora that has increasingly become available has shifted the focus from creating accurate digital texts to sophisticated digital editions. Previously prefaces, introductions, indexes, bibliographies, notes, critical apparatus (usually at the end of the page, in footnote size), and textual variations of different editions have either been discarded or systematically ignored in the creation of early digital collections. The ancient text that we read in modern editions, however, is the product of editors' choices, where they have evaluated the most probable variants attested in the manuscripts or the best conjectures provided by previous scholars. Humanists thus need both textual and paratextual information when they deal with ancient works.

Critical editions of classics are challenging for OCR systems in many ways. First, the layout is divided into several text flows with different font sizes: the author's text established by the editor, the critical apparatus where manuscript variants and scholars' conjectures are registered and, optionally, boxes for notes or side by side pages for the parallel translation. Second, ancient Greek utilizes a wide set of characters to represent the combinations of accents and breathing marks on the vowels, which are error prone for OCR systems. Third, critical editions are typically multilingual, because the critical apparatus is usually in Latin, names of cited scholars are spelled in English, German, French, Italian or other modern languages, and the prefaces, introductions, translations and indexes are also often in Latin or in modern languages. Finally, 19th century and early 20th century editions can have many damaged text pages that present great difficulties for conventional OCR.

2 Related Work

We can divide works related to the digitization of ancient texts into three groups: the first one concerns the analysis of manuscripts and early printed editions, the second group concerns the structure of digital critical editions (i.e. editions that register variants and conjectures to the established text) and the third group concerns OCR work performed on printed critical editions from the last two centuries.

The general approach for the first group is to provide methods and tools for computer assisted analysis and correction. Moalla et al. [17] developed a method to classify medieval manuscripts by different scripts in order to assist paleographers. Ben Jlaiel et al. [4] suggested a strategy to discriminate Arabic and Latin modern scripts that can be applied also to ancient scripts. Leydier et al. [14], [15] and Le Bourgeois et al. [13] used a method of word-spotting to retrieve similar images related to hand written words contained in manuscripts. Edwards et al. [10], on the other hand, developed a method based on a generalized Hidden Markov Model that improved accuracy on Latin manuscripts up to 75%.

The second group of studies explored recording variants and conjectures of modern authors, for instance Cervantes, such as Monroy et al. [18] or of ancient texts, for instance in Sanskrit, such as Csernel and Patte [9].

The third group of studies concerned improvements of OCR accuracy through post-processing techniques on the output of a single or multiple OCR engines. Ringlstetter et al. [27] suggested a method to discriminate character confusions in multilingual texts. Cecotti et al. [6] and Lund and Ringger [16] aligned multiple OCR outputs and illustrated strategies for selection. Namboodiri et al. [20] and Zhuang and Zhu [32] integrated multi-knowledge with the OCR output in post-processing, such as fixed poetical structures for Indian poetry or semantic lexicons for Chinese texts.

This paper further develops some guidelines first expressed in Stewart et al. [30]. In this previous research, the recognition of Greek accents in modern editions was not considered due to the technological limitations imposed by the OCR systems available.

3 Methodology

Our main interest in this research is to establish a work-flow for the massive digitization of Greek and Latin printed editions, with particular attention to the scalability of the process. The principal factors that determine the preparation of different pre- and postprocessing procedures are book collection specificities and preservation status.

3.1 Texts

Our experiments have been performed on different typologies of samples, in order to combine the aforementioned factors. Three editions of Athenaeus' *Deip-*

nosophistae and one of Aeschylus' tragedies have been used, by randomly extracting five pages from each exemplar. All documents have been downloaded from [12]. Athenaeus' exemplars belong to different collections and they are distributed along two centuries: Meineke's (1858) and Kaibel's (1887) editions are in the Teubner classical collection, whereas Gulick's (1951) second edition is in the Loeb classical library. Teubner and Loeb editions sensibly differ for script fonts, so that two different training sets have been created. They differ also for content organization: Meineke has no critical apparatus, Kaibel has a rich apparatus and Gulick has a minimal critical apparatus, supplementary notes and an English translation side by side.

The posthumous Hermann's (1852) edition of Aeschylus, published by Weidmann, has no critical apparatus and has a script very similar to the Teubner editions.

In this study, Greek text and critical apparatus have been separated manually, whereas English translation and notes have been ignored. In a second stage of the work, simple heuristics will be applied to classify textual areas.

Finally, in order to evaluate if and how the system could be extended to very early printed editions, an experiment has been performed on the *incunabulum* of Augustinus' *De Civitate Dei*, Venetiis 1475. In this case, even if the quality of the image is good, the irregularity of the script and the use of ligatures and abbreviations is very challenging.

3.2 OCR engines suitable for Ancient Greek recognition

Three OCR engines have been employed: Ideatech Anagnostis 4.1, Abbyy FineReader 9.0 and OCRopus 0.3 in bundle with Tesseract 2.03.

Anagnostis [2] is the unique commercial OCR engine that is provided with built-in functionality for ancient Greek and it can also be trained with new fonts. Accents and breathing marks are processed separately from the character body, improving the precision of the recognition system. On the other hand, Anagnostis is not able to recognize sequences of polytonic Greek and Latin characters, such as are present in the critical apparatus. In this case, Latin characters are rendered with the Greek characters most similar in shape (for example, the Latin letter *v* is transformed into the Greek letter *ν*).

FineReader [1] is capable of complex layout analysis and multilingual recognition. Even if polytonic Greek is not implemented natively, it is possible to train FineReader with new scripts, associating the images of glyphs to their Unicode representations. For these reasons, FineReader is currently the most reliable engine to recognize texts where different character sets are mixed.

OCRopus [22] is an open source project hosted by Google Code, that can be used in bundle with Tesseract [31], illustrated by Smith [28], which is one of the most accurate open source OCR engines currently available. OCRopus/Tesseract needs to be trained in order to recognize polytonic Greek (or other new scripts, except Latin scripts) and the recognition of mixed character sets is acceptable. The output format is plain text or xhtml enriched with a microformat to register positions of words (or optionally single characters) on the page image.

3.3 Training of single engines

The training process is divided into two phases. First, each OCR engine has been trained with pages randomly selected from the editions used in the experiments, verifying that the training set had no overlappings with the test set. Anagnostis and FineReader have been trained with the same sets of pages, whereas OCRopus/Tesseract has been trained with a different set, in order to increase the possibility of capturing character samples ignored by the other engines. In fact, the major issue in training FineReader and OCRopus/Tesseract with ancient Greek is caused by the high number of low frequency characters (according to the Zipfian law). Unicode represents polytonic Greek both by pre-combined characters and combining diacritics, but during the training process these engines seem to analyze glyphs only as whole characters, without separation between vowels and diacritics, as Anagnostis is able to do. The entire set of pre-combined characters for ancient Greek contains more than two hundred glyphs, but some of them are employed with a very low frequency. For example, in the Athenaeus' Kaibel edition, letter ᾷ (alpha with circumflex accent, rough breathing mark and iota subscript) occurs only twice out of more than one million characters. Thus, the probability that these rare characters are sampled in the training sets is quite low. As stated above, training is based on collections and not on exemplars, for the sake of scalability. For this reason, only one training set per engine has been created for the Teubner editions, mixing pages from both Kaibel's and Meineke's exemplars.

FineReader has a good built-in training set for modern (monotonic) Greek and it is possible to use the user defined training sets either alone or in bundle with the built-in trainings. Unfortunately, while this increases the accuracy for the recognition of non-accented characters it also decreases the accuracy for the recognition of vowels with accents and breathing marks. Thus, two training sets have been created for FineReader: with and without the addition of built-in training sets.

Second, the errors produced by each engine after the first stage have been compared with the ground truth, in order to calculate the error patterns that can be corrected by the cooperation of different OCR engines. The new training sets must be identical for all the engines. For Weidmann's edition, a new set of five pages, different from both the training set and the test set, has been recognized and the hand transcription has been used as ground truth. For the other editions, a k-fold cross validation method has been performed, using all the pages but the testing one for the training.

OCR output has been post-processed with a script that adjusts encoding and formatting errors, such as Latin characters inside Greek words with the same or very similar shape (e.g. Latin character *o* and Greek character ο, omicron), spaces followed by punctuation marks and other illegal sequences. A second script adjusts a small set of very frequent errors by the application of regular expressions. For example, a space followed by an accented vowel and by a consonant, an illegal sequence in ancient Greek, is transformed into space, followed by a vowel with breathing mark and a consonant.

The adjusted OCR output has been aligned to the ground truth by a dynamic programming alignment algorithm, according to the methods explained in Feng and Manmatha [11] and in van Beusekom et al. [5]. As usual, alignments are performed minimizing the costs to transform one string into the other, adding gap signs when it is necessary. In this way, n-gram alignments can be a couple of identical items (correct output), a couple of different items (error by substitution), an item aligned to a gap sign (error by insertion) or, finally, a gap sign aligned to an item (error by deletion). After the alignment, the average number of substitutions, insertions and deletions has been used to compute the average accuracy of each OCR engine. [21] offers a survey on methods to calculate approximate string matchings.

Data concerning alignments of single characters, bigrams, trigrams and tetragrams are registered in the error pattern file. For the sake of efficiency, data related to correct alignments of n-grams are registered only if the n-gram occurs at least once in a misalignment. In fact, we are particularly interested in comparing the probability that one n-gram is wrong to the probabilty that it is correct, as we will see below. The error pattern file is a table with four columns: number of characters the n-gram is constituted by, n-gram in OCR output, aligned n-gram in ground truth and a probability value, illustrated by formula (1).

$$\frac{C(a \rightarrow b)}{C(b)} * \left(\frac{C(b)}{N}\right)^{1/3} \tag{1}$$

The first factor of this value expresses the probability that, given a character (or n-gram) a in the OCR output, it represents a character (or n-gram) b in the ground truth (a is equal to b, in case of correct recognition). It is represented by the number of occurrences of the current alignment, $C(a \rightarrow b)$, divided by the total number of occurrences of the b character (or n-gram) in the ground truth, $C(b)$. The second factor of this value is the cubic root of $C(b)$ divided by the total number of characters or n-grams, N. This factor is equal for every engine, because it is based only on ground truth. The cubic root of this value is provided, according to the formula (6), which will be explained below.

3.4 Multiple Alignment and Naive Bayes Classifier

Tests have been performed on each OCR engine and the output has been adjusted with the simple post-processing scripts used also for the training samples. First of all, the two FineReader outputs (with and without the built-in trainings) have been aligned with the same methodology explained below for the alignments among different engines and we have obtained a new, more accurate FineReader output to be aligned with the other engines.

Outputs of the three engines have been aligned by a progressive multiple sequence alignment algorithm, as illustrated in Spencer [29]. The general principle of progressive alignment is that the most similar sequence pairs are aligned first, necessary gaps to align the sequences are fixed and supplementary gaps (with minimal costs) are progressively added to the previous aligned sequences, in order to perform the total alignment. In order to establish which pairs are

more similar and then must be aligned first, a phylogenetic tree should be constructed, but for our triple alignment it is enough to rate each engine according to the average accuracy value established during the training process. In our tests, FineReader has scored the highest, followed by OCRopus and Anagnostis. For this reason, FineReader and Anagnostis are aligned first. The resulting OCRopus string with gap signs is aligned to Anagnostis and the new gap signs are propagated to the previously aligned FineReader string. The triple alignment is shown in Figure 1, where the gap sign is represented by underscore.

The alignment in itself is not enough to determine the most probable character: even if two engines are in agreement, but are poorly reliable for a specific character identification, the most probable character could be provided by the third engine in disagreement. Even if all the engines are in agreement, the most probable character could be another one, such as when three engines are only able to recognize Greek characters and the text is written in Latin. This situation, however, is not considered in the current study, which is limited to the selection among characters provided by at least one engine.

Formally, the probability that the current position in the original printed page e_0 contains the character x, given that the first engine e_1 provides the character c_1, the second engine e_2 provides the character c_2 and the third engine e_3 provides the character c_3, is expressed by the formula:

$$P(e_0 = x | e_1 = c_1, e_2 = c_2, e_3 = c_3) \tag{2}$$

where, in general, $P(E_0|E_1, E_2, E_3)$, denotes the posterior probability for the event E_0, given the conjunction of the events $E_1 \cap E_2 \cap E_3$.

For example, (2) expresses the probability that the character $\acute{\alpha}$ is in the current position on the printed page, knowing that the first engine has provided $\dot{\alpha}$, the second engine has provided $\ddot{\alpha}$ and the third engine has provided $\acute{\alpha}$. These probabilities are deduced by the error pattern data recorded during the training process.

To find the highest probability among the three items provided by the engines, we have implemented a naive Bayes classifier. In virtue of the Bayes' theorem, from (2) follows:

$$[P(e_1 = c_1, e_2 = c_2, e_3 = c_3 | e_0 = x) * P(e_0 = x)]/P(e_1 = c_1, e_2 = c_2, e_3 = c_3) \tag{3}$$

Given that a naive Bayes classifier is based on the conditional independence assumption, the first factor in the numerator of (3) can be rewritten as

$$P(e_1 = c_1 | e_0 = x) * P(e_2 = c_2 | e_0 = x) * P(e_3 = c_3 | e_0 = x) \tag{4}$$

Considering that we are not interested in finding the value of the highest probability, but simply in finding the argument x_0 that provides the highest probability, we can omit the denominator of (3) and use the following formula:

$$x_0 = argmax\, P(e_1 = c_1 | e_0 = x) * P(e_2 = c_2 | e_0 = x) * P(e_3 = c_3 | e_0 = x) * P(e_0 = x) \tag{5}$$

Generalizing, we can write the equation (5) as

$$x_0 = argmax \prod_{i=1}^{n} P(e_i = c_i | e_0 = x) * P(e_0 = x)^{1/n} \tag{6}$$

where n is the number of OCR engines, e_i is a specific engine, c_i is the character provided by that engine. This equation explains why we computed the cubic root of the ground truth character probability in the equation (1). For the sake of efficiency, in this way we do not need to search for this factor and multiply it for the other factors all the times that we compute the requested term.

In our implementation, a triple agreement is unprocessed and in case of probability equal to zero, the output of the first engine (FineReader, in this case) is selected. In Figure 1 the result of the selection performed by the system is shown. In blue and red are indicated the correct characters selected from OCRopus and Anagnostis, despite the character recognized by FineReader.

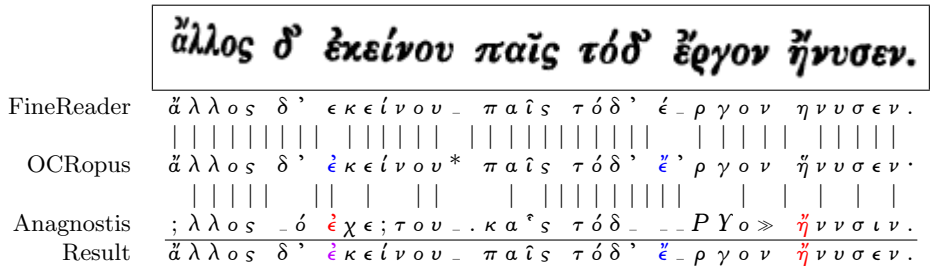

Fig. 1. Multiple alignment of the three engines output

3.5 Spell-checking supported by multiple alignment evidence

As stated above, the high number of ancient Greek pre-combined characters reduces the probability that the training sets contain some error patterns present in the test sets. In this case, the probability for a correct item is zero. On the other hand, as explained in Reynaert [25] and Stewart et al. [30], the automatic spell-checking applied to mispelled words alone is often unreliable; the first suggestion provided by the spell-checker could be wrong or, as is often the case, the word list of the spell checker does not contain proper names and morphological variants, and it thus replaces a correct word with an error. In order to reduce these issues, we have adopted a spell-checking procedure supported by the engines output evidence, filtering only the spell-checker suggestions that match a regular expression based on the triple alignment.

In order to integrate the spell-checker in our system, we have used the Aspell API [3] and we have used the word list generated by Morpheus, the ancient Greek morphological analyzer [7]. The string generated by the naive Bayes classifier is analyzed by the spell-checker. When words are rejected by the spell-checker because they are not contained in the word list, a regular expression is generated from the aligned original outputs, according to these simple rules: a) characters in agreement are written just once; b) two or three characters in disagreement are written between brackets; c) gaps are tranformed into question

marks (to indicate in the regular expression that the previous character or couple of characters between brackets are optional). For example, given the aligned outputs: a) *ἤλασεν*, b) *ἤλαστν* and c) *ἤλασ_ν*, the regular expression generated is /[*ἠἤ*]*λασ*[*ετ*]?*ν*/. All the suggestions provided by the spell-checker are matched with this regular expression, and only the first one that matches is selected, otherwise the mispelled word is left unchanged. Further examples are shown in Figure 2. The first example, *ἐξερήμωσεν*, and the last example, *ευφρων*, merit some further consideration. The first case reflects when a correct morphological variant is not present in the spell-checker word list. No suggestion provided by the spell-checker matches the regular expression generated by aligned outputs, thus the word is correctly left unchanged. On the other hand, *ευφρων* is an incorrect ancient Greek word because it has neither accent nor breathing mark. In this case, none of the suggestions of the spell checker are supported by the aligned outputs evidence, thus in this case the word is incorrectly left unchanged. While the first suggestion of the spell-checker is incorrect, the third one is correct.

FineReader output	RegEx matching all OCRs	Spell-checker suggestions	Result
ἐξερήμωσεν	*ἐξερήἑ*?[*μι*]*ωσεν*	*ἐξερήμωσε*, *ἐξερήμωσέ*, *ἐξηρήμωσεν*	*ἐξερήμωσεν*
ωπασεν	[*ωοὥ*]*π*[*αο*]*σ*[*εό*]*ν*	*ὥπασεν*, *ὥπασέν*, *σπάσεν*	*ὥπασεν*
εν’	[*εἕ*]*ν*’	*ἐν*, *ἐν*’ ... *ἕν*’ (34th item)	*ἕν*’
επάσης	*ε*?*ά*?*πάσης*	*πάσης*, *πάσῃς* ... *ἁπάσης* (11th item)	*ἁπάσης*
ἐὐθυντὴριον	[*εἐ*][*ὐυ*]*θυντ*[*ὴή*]*ριον*	*εὐθυντήριον*, *εὐθυντήριόν*, *εὐθυντῆρι*	*εὐθυντήριον*
πρώτος	*πρ*[*ώῶ*]*τος*	*πρῶτος*, *πρῶτός*, *πρωτὸς*	*πρῶτος*
Κύρος	[*KXH*][*ύῦῖ*]*ρος*	*Κῦρος*, *Κῦρός*, *Κύπρος*	*Κῦρος*
εθηκε	[*εἐ*]*θηκε*	*ἔθηκε*, *ἔθεκέ*, *θῆκε*	*ἔθηκε*
Δυδῶν	[*ΔΛ*]*υδῶν*	*Δυῶν*, *Διδῶν* ... *Λυδῶν* (6th item)	*Λυδῶν*
λάὸν	*λ*[*αά*][*ὸό*]*ν*	*λαὸν*, *λαόν*, *Λάιόν*	*λαὸν*
ἤλασεν	[*ἠἤ*]*λασ*[*ετ*]?*ν*	*ἤλασεν*, *ἤλασέν*, *ἤασεν*	*ἤλασεν*
ευφρων	*ε*?*ι*?[*υὄ*]*φρωο*?*ν*	*ἐύφρων*, *Εὔφρων*, *εὔφρων* (correct)	*ευφρων*

Fig. 2. Spell-checking supported by OCR evidence

3.6 The last test on a Latin *incunabulum*

The last test has been performed using a singular engine, OCRopus, on Augustinus’ *De Civitate Dei*, Venetiis 1475. We were interested in training OCRopus with Latin abbreviations and ligatures, encoded in Unicode according to the Medieval Unicode Font Initiative (MUFI) directions [19]. Images have been preprocessed with the OCRopus libraries for morphological operations, such as erosion and dilation and improvements due to preprocessing have been compared to ground truth.

4 Results

Results are evaluated comparing the accuracy of singular engines with the accuracy of the merged, spell-checked output. In order to compute the accuracy, the final output has been aligned with the ground truth. According to Reynaert [26], the accuracy has been calculated as:

$$\frac{matches}{matches + substitutions + insertions + deletions} \tag{7}$$

4.1 Accuracy of the single engines

Accuracy of single engines largely depends on the training sets created for each collection. Results are shown in Table 1. Both the most accurate OCR commercial application, Abbyy FineReader, and the most accurate OCR open source application, OCRopus/Tesseract are now provided with training sets that allow them to deal with polytonic Greek. In the case of Kaibel's exemplar, we have obtained better results with OCRopus/Tesseract than with Abbyy FineReader, suggesting that the open source software is currently mature enough to be applied to classical critical editions.

Results on Kaibel's and Meineke's exemplars, both Teubner editions, have been obtained using a single training set. The similarity of these results suggest that the project is scalable with pre-processing data reusable on exemplars of the same collections.

Edition	FR w/o built-in training	FR with built-in training	OCRopus	Anagnostis
Gulick (Loeb)	96.44%	94.35%	92.63%	93.15%
Kaibel (Teubner)	93.11%	93.15%	95.19%	92.97%
Meineke (Teubner)	94.54%	93.79%	92.88%	91.78%
Hermann (Weidmann)	97.41%	N/A	91.84%	78.64%

Table 1. Accuracy: single engines

4.2 Improvements due to alignment and constrained spell-checking

Improvements due to alignment can be divided in two steps. In fact, the first gain is due to the alignment of the FineReader outputs, with and without the built-in training set, in cooperation with the user training set. In average, the improvement is +1.15% in relation to the best single engine, which is FineReader without the built-in training except in the case of Kaibel, as stated in the previous section.

The second step is the triple alignment and constrained spell-checking, which provides a gain, in average, of +2.49% in relation to the best single engine. A t-test for each exemplar demonstrates that improvements are always significant, with $p<0.05$. Analytical results are provided in Table 2.

The best result, as expected, concerns the most recent Loeb edition, with an accuracy rate of 99.01%. If we consider only the case insensitive text (without punctuation marks, breathing marks and accents), the accuracy arises to 99.48%. This value is especially important if we are interested in evaluating the expected recall of a text retrieval system, where ancient Greek words can be searched in upper case.

Edition	Alignment and spell-checking	Aligned FR	Best engine
Gulick (Loeb)	99.01%	98.02%	96.44%
gain	+2.57%	+1.58%	0.00%
Kaibel (Teubner)	98.17%	95.45%	95.19%
gain	+2.98%	+0.26%	0.0%
Meineke (Teubner)	97.46%	96.15%	94.54%
gain	+2.92%	+1.61%	0.00%
Hermann (Weidmann)	98.91%	N/A	97.41%
gain	+1.50%	N/A	0.00%

Table 2. Accuracy: alignment and spell-checking

4.3 Accuracy on the critical apparatus

Tests on the critical apparatus of Gulick's and Kaibel's editions have been performed without a specific training for the footnote size, but with the same training sets applied to the rest of the page. Only the FineReader output with the built-in training set has been used, because the output created without it had a very low accuracy.

The average accuracy due to the triple alignment is 92.01%, with an average gain of +3.26% in relation to the best single engine, that is FineReader on Gulick's edition and OCRopus/Tesseract on Kaibel's edition. Analytical results are provided in Table 3. Also on the critical apparatus, t-test demonstrates that improvements are significant, with $p<0.05$.

It is important to point out that the critical apparatus, according to estimations computed in Stewart et al. [30], is approximately, on average, 5% of the page in editions with minimal information (such as Loeb editions), and 14% of the page, on average, for more informative apparatus (such Teubner editions).

	Alignment and spell-checking	FR with b.-in	OCRopus	Anagnostis
Gulick	90.88%	87.99%	64.79%	59.08%
gain	+2.89%	0.0%	-23.20%	-28.91%
Kaibel	93.14%	87.68%	89.54%	57.11%
gain	+3.60%	-1.86%	0.0%	-32.43%

Table 3. Accuracy: critical apparatus

4.4 Accuracy on the *incunabulum*

The test performed with OCRopus on Augustinus' *De Civitate Dei* provides an accuracy of 81.05%, confirming results reached by Reddy and Crane [24].

5 Conclusion

The software developed for this study and the updated benchmarks are available on the Perseus Project website [23].

As claimed in Crane et al. [8], in order to go beyond digital incunabula it is necessary to build a digital library of classical critical editions, on which information extraction, natural language processing and corpus analysis techniques should be perfomed. A satisfactory OCR accuracy rate for the whole content of a critical edition (text and apparatus), that will allow us to lower the costs for post-corrections by hand, is one first necessary step to build the new generation of textual corpora.

6 Acknowledgments

This work was supported by a grant from the Mellon Foundation. We also gratefully acknowledge Marco Baroni and Tommaso Mastropasqua, of CIMeC - University of Trento (Italy), for their useful suggestions.

References

1. Abbyy FineReader Homepage, http://www.abbyy.com
2. Anagnostis Homepage, http://www.ideatech-online.com
3. Aspell Spell-checker Homepage, http://aspell.net
4. M. Ben Jlaiel, S. Kanoun, A.M Alimi, R. Mullot: Three decision levels strategy for Arabic and Latin texts differentiation in printed and handwritten natures. 9th International Conference on Document Analysis and Recognition, 1103–1107 (2007)
5. J. van Beusekom, F. Shafait, T.M. Breul: Automated OCR Ground Truth Generation. 9th International Conference on Document Analysis and Recognition, 111–117 (2007)
6. H. Cecotti, A. Belaïd: Hybrid OCR combination approach complemented by a specialized ICR applied on ancient documents. 8th International Conference on Document Analysis and Recognition, 1045-1049 (2005)
7. G. Crane: Generating and parsing classical Greek. Literary and Linguistic Computing, 6, 4, 243–245 (1991)
8. G. Crane, D. Bamman, L. Cerrato, A. Jones, D. Mimno, A. Packel, D. Sculley, G. Weaver: Beyond Digital Incunabula: Modeling the Next Generation of Digital Libraries. 10th European Conference on Research and Advanced Technology for Digital Libraries, volume 4172 of Lecture Notes in Computer Science, 353-366, Springer (2006)
9. M. Csernel, F. Patte: Critical Edition of Sanskrit Texts. 1st International Sanskrit Computational Linguistics Symposium, 95–113 (2007)

10. J. Edwards, Y.W. Teh, D. Forsyth, R. Bock, M. Maire, G. Vesom: Making Latin Manuscripts Searchable using gHMM's. Advances in Neural Information Processing Systems, 17, 385—392. (2004)
11. S. Feng, R. Manmatha: A Hierarchical, HMM-based Automatic Evaluation of OCR Accuracy for a Digital Library of Books. JCDL 2006, 109–118 (2006)
12. Internet Archive Homepage, http://www.archive.org
13. F. Le Bourgeois, H. Emptoz: DEBORA: Digital AccEss to BOoks of the RenAissance. International Journal on Document Analysis and Recognition 9, 192–221 (2007)
14. Y. Leydier, F. Lebourgeois, H. Emptoz: Text search for medieval manuscript images. Pattern Recognition, 40, 12, 3552–3567 (2007)
15. Y. Leydier, F. Le Bourgeois, H. Emptoz: Textual Indexation of Ancient Documents. 2005 ACM symposium on Document engineering, 111–117 (2005)
16. W.B. Lund, E.K. Ringger: Improving Optical Character Recognition through Efficient Multiple System Alignment. (to appear in JCDL 2009)
17. I. Moalla, F. Lebourgeois, H. Emptoz, A.M. Alimi: Image Analysis for Paleography Inspection. Document Analysis Systems VII, 25–37 (2006)
18. C. Monroy, R. Kochumman, R. Furuta, E. Urbina, E. Melgoza, A. Goenka: Visualization of Variants in Textual Collations to Analyze the Evolution of Literary Works in The Cervantes Project. 6th European Conference on Research and Advanced Technology for Digital Libraries, 638–653 (2007).
19. Medieval Unicode Font Initiative Homepage, http://www.mufi.info/fonts
20. A.M. Namboodiri, P.J. Narayanan, C.V. Jawahar: On Using Classical Poetry Structure for Indian Language Post-Processing. 9th International Conference on Document Analysis and Recognition - Volume 02, IEEE Computer Society, 1238–1242 (2007)
21. G. Navarro: A Guided Tour to Approximate String Matching. ACM Computing Surveys, 33, 1, 31–88 (2001)
22. OCRopus Homepage, code.google.com/p/ocropus
23. Perseus Project Homepage, http://www.perseus.tufts.edu/hopper/opensource
24. S. Reddy, G. Crane: A Document Recognition System for Early Modern Latin. Chicago Colloquium on Digital Humanities and Computer Science: What Do You Do With A Million Books, Chicago, IL (2006).
25. M. Reynaert: Non-interactive OCR Post-correction for Giga-Scale Digitization Projects. A. Gelbukh (ed.): CICLing 2008, LNCS 4919, 617–630 (2008)
26. M. Reynaert: All, and only, the Errors: more Complete and Consistent Spelling and OCR-Error Correction Evaluation. 6th International Conference on Language Resources and Evaluation 2008, 1867–1872 (2008)
27. C. Ringlstetter, K. Schulz, S. Mihov, K. Louka: The same is not the same - post-correction of alphabet confusion errors in mixed-alphabet OCR recognition. 8th International Conference on Document Analysis and Recognition, 1, 406–410 (2005)
28. R. Smith: An Overview of the Tesseract OCR Engine. 9th International Conference on Document Analysis and Recognition, 2, IEEE Computer Society, 629–633 (2007)
29. M. Spencer, C. Howe: Collating texts using progressive multiple alignment. Computer and the Humanities, 37, 1, 97–109 (2003)
30. G. Stewart, G. Crane, A. Babeu: A New Generation of Textual Corpora. JCDL 2007, 356–365 (2007)
31. Tesseract Homepage, http://code.google.com/p/tesseract-ocr
32. L. Zhuang, X. Zhu: An OCR Post-processing Approach Based on Multi-knowledge. 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, 346–352 (2005)