A generic approach for OCR performance evaluation

A. Belaïd and L. Pierron

LORIA-CNRS/INRIA
Campus scientifique BP 239
54500 Vandœuvre-lès-Nancy, FRANCE
email : {Laurent.Pierron,Abdel.Belaid}@loria.fr

1 Introduction

For different document automation operations it is always needed to have an OCR evaluation phase to select the most interesting OCRs for the document class studied. The evaluation should indicate the defects and drawbacks of each OCR and allow to determine the required heuristics to combine these OCRs in order to obtain the highest performances in production: the lowest reject rate for a predefined confusion rate (in general 1/10000). The evaluation should be done automatically and completely integrated in a more global OCR platform.

In this paper, we present our experience in OCR evaluation for four kind of commercial OCRs on few document classes. We will comment the results, the methodology used, the encountered problems and propose some heuristics to improve these results.

2 Global Methodology

Figure 1 shows the two main parts of the industrial OCR process proposed in this study. The first part is related to the evaluation OCR. The result of this part is used as a customization of the production phase. The production phase consists in combining several OCRs for document retro-conversion of thousands pages with the constraint to attempt an error rate less than 1/10 0000, with the lowest possible correction human effort.

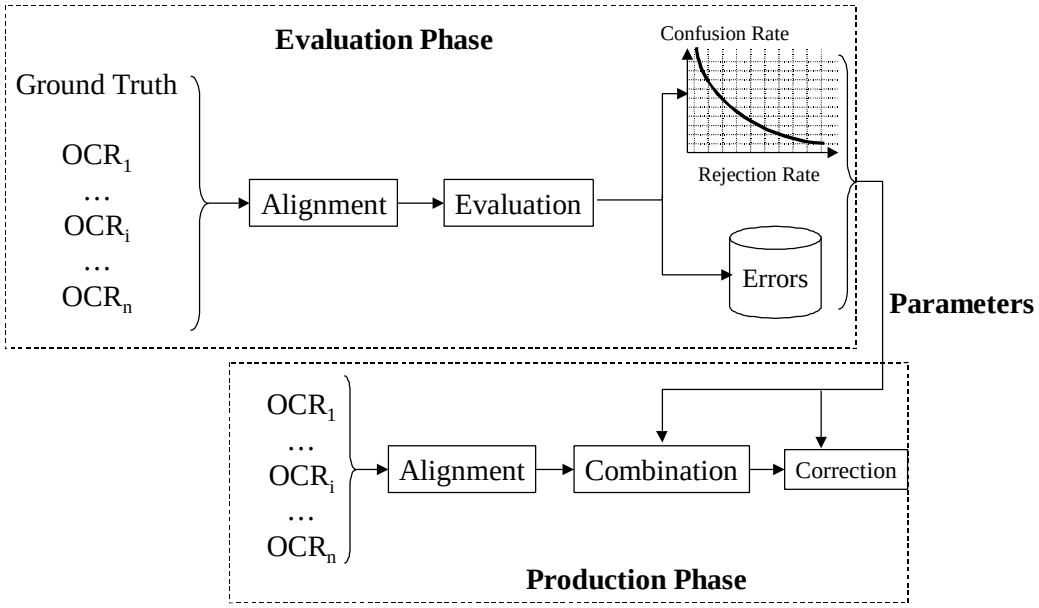

Figure 1: Evaluation Modules

The evaluation phase consists of comparing an OCR result sample (OCR_1, …, OCR_i, …, OCR_n) with a Ground Truth (GT) and producing confusion-reject graphs and a list of residual errors. The document samples correspond to a part of the mass of documents to be processed. The evaluation phase is divided into two steps: the alignment step which synchronizes the corresponding files, and the evaluation step which extracts the graphs and the errors from the synchronization process.

The production phase combines the OCR (OCR_1, …, OCR_i, …, OCR_n) results by using the parameters determined by the evaluation phase. The production phase is divided into three different steps: the alignment step which synchronizes the OCR results, the combination step which selects the most probable local OCR response (characters) by associating it a confidence score. Human correction is operated at the end of the process to control characters which confidence scores are less than a given failure score. The failure score is determined in the evaluation phase in order to obtain an error rate after correction less than 1/ 10 000.

3 OCR Evaluation

3.1 Ground Truth

3.1.1 Representativeness

The may correspond to a very representative sample of the class document studied. This means that the GT may contain the same character frequency and the same graphical attributes frequency that ones in the original document. Furthermore the sample should be enough long to allow statistics computing. Statistically speaking, to be sure to attempt the error rate less than 1/10 000, it is required to have at minimum 20 000 samples per character class (The Moore law imposes to double the sampling in order to measure a frequency).

Without revealing the document origin because of the confidential aspect of this work, 138 pages have been used. The document is composed of two columns, typed in very small size font. The total character number is equal to 592 187 but within each class the character number vary from 1 (for some rare lett ers) to 85 836 for spaces. In this conditions, we remark that we cannot sub-evaluate or up-evaluate the performances for a majority of character classes where the account number is less than 5 000. Only the recognition of these letters mentioned in the following table can be operated with the desired precision.

a	32658	l	22 279	s	37 091
d	20 695	n	31 555	t	36 010
e	61 385	o	24 026	u	23 795
i	33 883	r	32 144	space	85836

3.1.2 Format

The GT is seen as a list of consecutive character lines given in the same sequence that one produced by OCRs. This means that if the document structure is complex (i.e. hierarchical or mosaic), it is ordered into a linear structure where a zone composed of a list of lines is followed by another zone. It is obvious that the GT may have the same order.
The GT can be produced in three different ways:

- From existing text, this poses a lot of problems because the text is often interpreted leading to some word changing (for example, the date is converted in a specific format);
- Manually, by typing and checking the text. This needs very strict typing orders such as how to type hyphens, indexes, exponents, etc.
- Produced by OCR and corrected by a human operator. This is the most reliable method because the result corresponds exactly to the OCR result format.

In all cases, a rendition is made on the image to create the GT. As in the third case, this is the OCR that makes the rendition and as there are the OCR results that we want to compare to the GT, the image rendition is the same. In our recent experimentation, we have used the three methods and remarked that the third method is that gave the best result.

3.2 OCR alignment

The objective of the alignment process is to highlight the different error cases such as insertion, deletion and substitution. These errors will constitute the basis for the correction algorithms. Several alignment algorithms exist in the literature based either on dynamic programming or on the longest common string [5-9]. The latter can generate some defects if there exist in the text a repetition of some similar sub-strings. So, for this reason, we have used the Myers's algorithm based on an optimal dynamic programming matching. However, this algorithm, used also by GNU diff, disfavors the substitution which sometimes gives an unnatural synchronization, but it is so rare (1 / 10 000 characters) that we have decided to keep it.

To compare more than two files, the application of this algorithm is exponential. So we have proposed an iterative approach allowing the comparison on pairs. A document reference is first chosen. Then, all the other documents are compared to this reference document one by one. At last, the comparison results are merged.

3.3 OCR evaluation

3.3.1 OCR engines

Four OCRs have been evaluated. They correspond to:

- *FineReader 4.0* of *Abby*
- *OmniPage* of *Caere DevKit 2000*
- *Recognita* of *Caere DevKit 2000*
- *TextBridge 4.5* of *ScanSoft*

For each OCR a program is performed which takes as arguments a file containing the image in TIFF and a list of processing options depending on each OCR and output options. The output format used for the study is the most complete one that can be given by an OCR. It is similar to one of the *DevKit 2000* output format. The output document is a textual document readable by a human. In this document each line represents a recognized character and each column an attribute associated to this character. The first document column contains always the recognized character.
This output format is proposed in standard by *OmniPage* and *Recognita*. For *FineReader*, the format performing was forthright because the recognition uses a similar table. *TextBridge* doesn't propose access to the internal recognition structure but proposes an output tagged named *XDOC* which contains some recognition information. We have written a program which reads the *XDOC* format and gives the recognition information in a table.

3.3.2 Graphs

We have used the previous table in order to perform the graphs giving the correspondence between confusion and rejection rates.

Let a_k be the number of well recognized characters and b_k be the number of bad recognized characters for the confidence coefficient k.

The rejection rate for the confidence coefficient k is equal to: $\frac{\sum_{i=0}^{i<k} a_i + b_i}{\sum_{j=0}^{j\le n} a_j + b_j}$, the rejection rate is equal to 0 for k=0, and n is the value of the highest confidence score.

The confusion rate for the confidence coefficient *k* is equal to: $\frac{\sum_{i=k}^{i\le n} b_i}{\sum_{j=0}^{j\le n} a_j + b_j}$, the rejection rate is equal to 0 for k=0, and n is the value of the highest confidence score.

We have achieved a module which takes as an input a list of documents resulting from the evaluation, and which performs for each character and confidence coefficient k the confusion rate and the rejection rate. As the output of this module, a file is constructed per character containing the graph points.

FineReader and *TextBridge* have leaded to the creation of confusion graphs in function of the rejection rate. It seems difficult to realize the same thing with *OmniPage* and *Recognita* by using the same method because the latest will have only two points on the graph (because the confidence score takes only two values). TextBridge associates a confidence coefficient at each result which allows to draw the graphs with many points. FineReader gives four confidence levels for each recognized character. Hence, we have drawn the graphs with the four points connected by strokes , sometimes we can find less than four points.

3.3.3 Error cataloguing and heuristic extraction

The error list has been obtained from the document synchronization of all the OCRs and the GT. In the synchronization document, we have extracted all the FineReader answers which are not similar to the reference

without taking into account the confidence in the answer. The list obtained contains 390 errors. This list is given in HTML accompanied by the other OCR answers and the part of the recognized image. All the HTML document lines are then analyzed by hand in order to create a catalogue.

We have classified the errors in four different categories:

1. NOISE: the presence of a spot or of a small stroke on the image have misled FineReader which have recognized a sign or a diacritic or another character; this led to 62 errors (15,9% of the errors, i.e. a bit more 1 / 10000 error). These errors correspond to essentially supplementary punctuation symbols and supplementary accents.
2. CONFUSION: there is apparently no visual defect on the image but FineReader has not identified the right character; this led to 272 errors (69,7% of the errors, i.e. 5/10 000 error). The most important confusions are: 1 for I (40), - for hyphen (31), " for ° (26) and I for l (20).
3. IMAGE : a part of the character is erased, FineReader cannot identify the right character; this led to 19 errors (4,9% of the total , i.e. a bit more 0.25 / 10000 error).
4. MICRO-SEGMENTATION: although there is no visual defect on the image the characters are bad segmented by the OCR. There are 37 errors (9,5% of the errors, i.e. a bit more 1 / 10000 error).

From this cataloguing, some heuristics are proposed for the rejection, in order to decrease the residual error rate for a given rejection threshold. An example of heuristic is that which proposes to reject all the doubloons II isolated. For 60 cases, the system allow to highlight 20 errors.
After the application of 6 heuristics additionally to the combination of the two OCRs, we attempt the fateful confusion rate of 1/10000 precisely 0,9/10 000 for a rejection rate of 8,43%.

3.4 OCR Combination

In the different studies realized by Rice[1-4], it is shown that about 50% of error is eliminated by the combination of OCRs (by classical methods like majority vote) which individual recognition rate is 97%. So, this gain can be attempt only in the case where errors come from OCR and not from the image and where the OCRs are of good quality. But when the recognition rate becomes higher, about 99,5%, the classical combination doesn't bring any substantial improvement. Thus, we have chosen to base the combination on the selection of the best OCR assisted by a complementary one. This combination is reinforced by the use of heuristics extracted from the cataloguing error phase.

For combination, we consider that each OCR gives as a result a list of characters c_i accompanied by their confidence score k_i. The combination algorithm is as follows:
Let the two OCRs be respectively named OCR' and OCR''

If $k_i^{'} >$ S' **then** select $c_i^{'}$ with coefficient $k_i^{'}$

else **if** $k_i^{''} >$ S'' **then** select $c_i^{''}$ with coefficient $k_i^{''}$

else character is rejected with confidence score equal to 0.

S' and S'' are the respecti ve rejection thresholds for OCR' and OCR''. The general idea is to supplement OCR' by OCR'' when OCR' doubts. It seems logical to choose the most efficient OCR as OCR'.

The two graphs given in Figure 2 show the confusion/rejection variations for FineReader in red, TextBridge in blue, and by applying the heuristics on FineReader in green, and in combining FineReader and TextBridge in black. The graph on the right part is a zoom of a left part of the left graph. We notice that all the TextBridge graph points are always higher than the FineReader graph points. So, we choose FineReader as the reference OCR for the combination approach and the heuristic application method. We remark too that the improvements obtained by only applying the heuristics, that do not need the use of TextBridge, is not so important as the one provided by the combination with TextBridge.

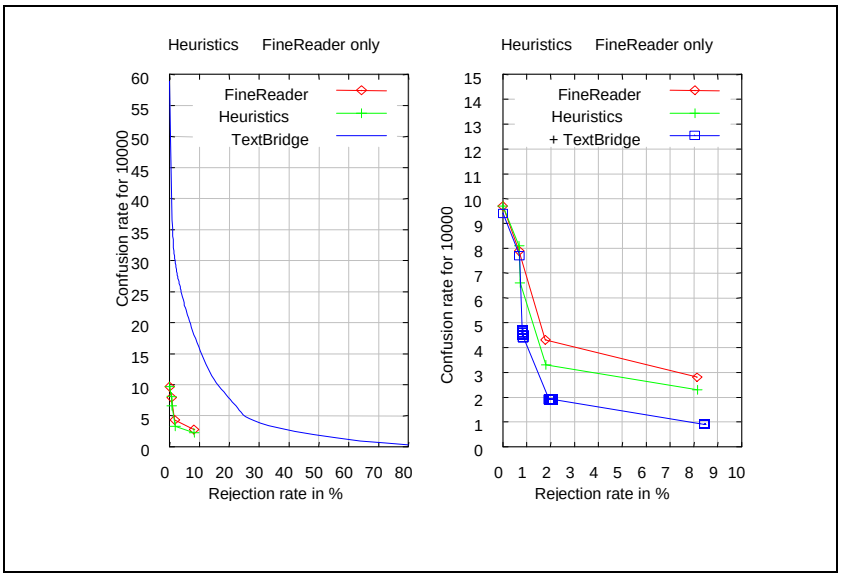

Figure 2 : Confusion/Rejection Graphs.

4 Conclusion

We have proposed a generic approach for OCR evaluation and combination. This approach has been validated in the framework of an industrial application for automatic document capture, by attempting the lowest score imposed of 1 error for 10 000 characters. The approach is based on the selection of a good OCR improved by a secondary OCR and heuristics. In the future we tend to apply this technique on other documents classes. The objective now is to decrease the rejection rate by using heuristics per document class and automatically performing error correction. Finally, we project to create a heuristic catalogue in order to be able to reuse them from one project to another and to quickly test the applicability of the heuristics on other projects.

5 References

[1] S. V. Rice, J. Kanai, and T. A. Nartker, "An Evaluation of OCR Accuracy," ISRI 1993 AnnuaI Research Report, University of Nevada, Las Vegas, April 1993, 9-31.

[2] S. V. Rice, J. Kanai, and T. A. Nartker, "The Third Annual Test of OCR Accuracy, " ISRI 1994 Annual Research Report, University of Nevada, Las Vegas, April 1994, 11-38.

[3] S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The Fourth AnnuaI Test of OCR Accuracy," ISRI 1995 Annual Research Report, University of Nevada, Las Vegas, April 1995, ll- 49.

[4] S. V. Rice, 'The OCR Experimental Environment, Version 3," ISRI 1993 Annual Research Report, University of Nevada, Las Vegas, April 1993, 83-86.

[5] R. A. Wagner and M. J. Fisher, "The String-to-String Correction Problem," Journal of the ACM, 21(l), (1974) 168-173.

[6] P. A. V. Hall and G. R. Dowling, "Approximate String Matching," ACM Comput. Surv. **12** (1980), 381-402.

[7] W. J. Masek and M. S. Paterson, "A Faster Algorithm Computing String Edit Machine," Journal Computer Systems Sci. **20** , 1 (1980) 18-31.

[8] C. K. Chow, "On Optimum Recognition Error and Reject Tradeoff," IEEE Transactions on Information Theory, Volume IT-16, No. 1, Jan. 1970, 41-46.

[9] W. Miller and E.W. Myers, A File Comparison Program, Software, Practice and Experience, Vol. 15, No. 11, p 1025-40.