ICDAR2017 Competition on Post-OCR Text Correction

Guillaume Chiron*, Antoine Doucet†, Mickaël Coustaty† and Jean-Philippe Moreux*

* National Library of France
Quai François Mauriac, 75706 Paris, France
(guillaume.chiron,jean-philippe.moreux)@bnf.fr

† L3i Lab, University of la Rochelle
Av. Michel Crépeau, 17000 La Rochelle, France
(mickael.coustaty,antoine.doucet)@univ-lr.fr

***Abstract*—This paper describes the ICDAR2017 competition on post-OCR text correction and presents the different methods submitted by the participants. OCR has been an active research field for over the past 30 years but results are still imperfect, especially for historical documents. The purpose of this competition is to compare and evaluate automatic approaches for correcting (denoising) OCR-ed texts. The challenge consists of two independent tasks: 1) error detection and 2) error correction. An original dataset of 12M OCR-ed symbols along with an aligned ground truth was provided to the participants with 80% of the dataset dedicated to the training and 20% to the evaluation. Different sources were aggregated and namely contain newspapers and monographs covering 2 languages (English and French). 11 teams submitted results, while the difficulty of the task was underlined by the fact that only half of the submitted methods were able to denoise the evaluation dataset on average. In any case, this competition, which counted 35 registrations, illustrates the strong interest of the community in this essential problem, which is key to any digitization process involving textual data.**

I. Introduction

The accuracy of Optical Character Recognition (OCR) technologies considerably impacts the way digital documents are indexed, accessed and exploited [1], [2]. During the last decades, OCR engines have been constantly improved and are able today to return exploitable results on mainstream documents. But in practice, digital libraries contain many transcriptions with a quality lower than expected. Ancient documents with challenging layouts and various levels of conservation such as historical newspapers still resist to modern OCR systems. Also, formerly digitized resources processed with outdated OCRs are rarely re-sent through the latest state-of-the-art digitization pipeline, as priority is often given to the ever-growing masses of newly incoming documents. Therefore, post-OCR approaches and benchmarks to evaluate the progress of the community in that field are more than ever needed.

Since 2003, more than 60 competitions have been organized in the different editions of ICDAR (2 in 2003, 3 in 2005, 3 in 2007, 9 in 2009, 17 in 2011, 17 in 2013 and 11 in 2015) but none of them were related to OCR post-correction approaches, although many techniques have been the subject of publication during these years [3], [4], [5], [6], [7].

In this context, the competition was open to researchers from several fields (document analysis, natural language processing, data analysis, text data mining, machine learning...) to challenge their method(s) for improving/denoising OCR-ed texts. The benefit is double as it gives a global overview of the methods developed by the community and it sets down a common baseline for further works.

An analysis of the state of the art shows that it remains difficult to find benchmarks to assess the performance of OCR correction algorithms. This competition focused on a newly released challenging dataset: 12 million characters in 2 languages. The dataset was distributed along with the metrics described in Section II-C and the corresponding evaluation script.

II. Competition setup

A. Tasks description

It has been decided to divide the challenge into 2 independent tasks (see Fig. 1), each focusing on a non-overlapping split of the dataset. This decision is meant to open the competition to a larger audience and also to include teams that only have a partial working system. Indeed, the possibility to bypass the first task can be supportive as the dataset is relatively noisy and thus could potentially lead to discouraging scores during the training phase. The two tasks of 1) detection and 2) correction are described below.

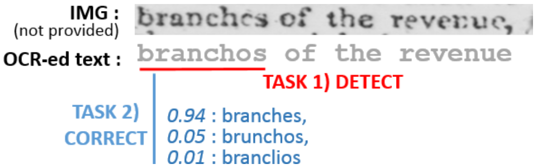


Fig. 1. Two tasks: error detection and error correction.

Task 1 - Detection of OCR errors: Given the raw OCR-ed text only, the participants are asked to provide the position and also the length of the suspected erroneous tokens. The length information is non-trivial; it is necessary in the case of words that are wrongly split (e.g. OCR-ed separators such as spaces, hyphens or line breaks).

Task 2 - Correction of OCR errors: Given the OCR errors in their context (position and length), the participants are asked to provide, for each error, a ranked list of replacement candidates (the list may contain only one). The ability to provide multiple candidates enables the evaluation of semi-automated techniques as we will detail later.

B. Dataset

The proposed dataset [1] has been built within the AMÉLIOCR project[1] on OCR post-correction. It accounts for 12M OCR-ed characters along with the corresponding Ground Truth (GT), with an equal share of English- and French-written documents (see Table I). The documents come from different digital collections available, among others, at the National Library of France (BnF) and the British Library (BL). The corresponding GT comes both from BnF's internal projects and external initiatives such as Gutenberg, Europeana Newspapers, IMPACT and Wikisource.

TABLE I
SOURCES, QUANTITIES AND AVERAGE CHARACTER ERROR RATES (E.R.) INVOLVED IN BOTH ENGLISH AND FRENCH PARTS OF THE DATASET.

Lang	Source	Type	Dates	E.R.	Char.
Eng.	BL Euro NP	serials	1744 - 1894	4%	1.8 M
	BL Monog	monog.	1858 - 1891	1%	1.2 M
	GT BnF Eng	monog.	1802 - 1911	2%	3.0 M
Fr.	Europeana NP	serials	1814 - 1944	4%	1.0 M
	IMPACT	monog.	1821 - 1864	1%	0.4 M
	GT BnF Fr	mixed	1686 - 1943	1%	2.0 M
	Digit. BnF	mixed	1654 - 2000	3%	0.2 M
	News other	serials	1897 - 1934	4%	0.6 M
	Monog other	monog.	1689 - 1883	3%	1.8 M

Total: 12 M

Degraded documents sometimes result in highly noisy OCR output and thus cannot reasonably be fully aligned with their GT. The unaligned sequences have not been included in the presented statistics (e.g. number of characters and error rates). Error rates vary according to the nature and the state of degradation of the documents. Historical newspapers for example, due to their complex layout and their original fonts have been reported to be especially challenging for OCR engines with up to 10% of wrongly detected characters in some documents.

A first part of the dataset (80%) was provided to the participants for training and testing purposes, and the rest (20%) has been used by the organizers for evaluation. Moreover, as the input of ”Task 2) correction” contains the answers of ”Task 1) detection”, each task was assigned to a different subset of the corpus. Fig. 2 illustrates on a sample file the format provided to the participants. Tokens are simply space-separated sequences, with no restriction on punctuation (examples of tokens: ”i”, ”i'am”, ”football?”, ”qm87-7lk.,qs'g”). Tokens that are considered miss-aligned with the GT are indicated by the ”#” signal. The ”@” signal is used as padding symbol in the aligned sequences.

C. Evaluation modalities

We proposed two different scenarii to assess the performance of the methods submitted by the participants.

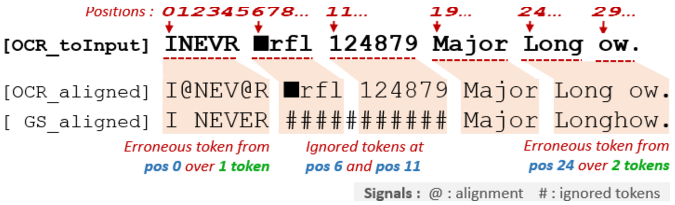

Positions : 012345678... 11... 19... 24... 29...
[OCR_toInput] INEVR ■rfl 124879 Major Long ow.
[OCR_aligned] I@NEV@R ■rfl 124879 Major Long ow.
[GS_aligned] I NEVER ########### Major Longhow.
Erroneous token from pos 0 over 1 token
Ignored tokens at pos 6 and pos 11
Erroneous token from pos 24 over 2 tokens
Signals : @ : alignment # : ignored tokens

Fig. 2. Sample of the training set provided to the participants.

Task 1: As it is purely a matter of tokens being truly erroneous or not, task 1 is evaluated with usual metrics: recall, precision and F-measure, the latter providing the official ranking of this task. The length information provided by the participants is automatically taken into account by default thanks to the alignment with the GT. For example, the two-token OCR error ”we ar” supposed to be ”wear” in the GT would penalize (regarding the recall) a solution with only the first token ”we”.

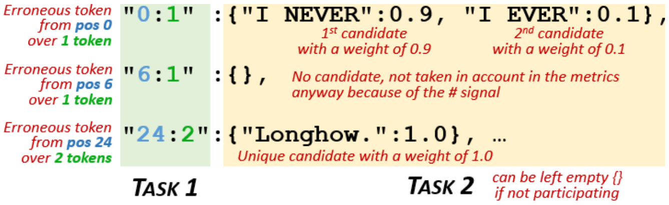
Task 2: The chosen metric for ranking considers for every token of the text sequence, a weighted sum of the Levenshtein distances between the correction candidates and the corresponding token in the Ground Truth. Consequently, best approaches are those that minimize this distance. Providing multiple candidates enables the evaluation on different modalities reflecting various scenarii. We proposed to focus on the 2 following:

- Fully automated scenario, meant for the comparative evaluation of fully automatic OCR correction tools, where only the top 1 (highest-weighted word) in each list is taken into account.
- Semi-automated scenario, meant for the comparative evaluation of human-assisted correction tools, where a person typically picks the right correction within a list of system-generated candidate corrections. Thus, it takes into account the list of proposed corrections along with their weights, with an arbitrary limitation to the top 6 candidates.

The evaluation script was made available to the participants during the competition (https://git.univ-lr.fr/gchiro01/icdar2017). It computes the metrics presented above over either the training set or the full dataset, with the assumption that the input files are correctly formated (see Fig. 3). The choice of using a structured format *key(pos,lenght) / value(candidates,weights)* rather than asking fully corrected sequences has been motivated by the bias that would have implied any further alignment process between the participants results and the corresponding GT.

Miss-aligned tokens (see ”#” signals) are ignored for the computation of the different metrics. Also, given the complexity of dealing with hyphen correction, it has been decided to ignore the tokens containing an hyphen through the evaluation. Thus, whether such errors are corrected does not impact on the final result.

[1] Led by the National Library of France (Department of preservation and conservation) and the L3i laboratory (Univ. of La Rochelle, France)



```
"0:1"  :{"I NEVER":0.9, "I EVER":0.1},
"6:1"  :{},
"24:2":{"Longhow.":1.0}, …
```

Erroneous token from pos 0 over 1 token
1st candidate with a weight of 0.9
2nd candidate with a weight of 0.1
Erroneous token from pos 6 over 1 token
No candidate, not taken in account in the metrics anyway because of the # signal
Erroneous token from pos 24 over 2 tokens
Unique candidate with a weight of 1.0
TASK 1
TASK 2
can be left empty {} if not participating

Fig. 3. Format expected for submissions to both Task 1 and 2.

D. *Modalities and timeline of the competition*

The competition was run in open mode (submission of the results but not the executables). We have relied on the scientific integrity of the authors to follow the rules of the competition. The authors were free to participate in one or both tasks, even on subparts of the dataset. The training set was made available mid-March 2017. The test set (without the GT), used for evaluating the different methods was made available on the 27th of June with 2 days given for the teams to submit their results.

III. SUBMITTED METHODS

In total, 35 teams registered to the competition, for a final number of 11 submissions. The following section gives a brief description of the submitted methods. The descriptions were provided by their authors and partially curated by the competition organizers (essentially for consistence and brevity).

5gram-KN-LV - Team from LIMSI[2]*, France*

This approach applied only to task 1, combines combines a standard 5-gram language model with Kneser-Ney smoothing with Levenshtein distance measures. The training material was first preprocessed (removal of signal symbols, cleaning of hyphens and punctuations variations) to obtain a clean text, which then served as a domain-specific lexicon for French and English respectively. These domain-specific lexicons were then combined with modern lexicons taken from the TICCL implementation, and served as candidates for recognized OCR errors. We also trained n-gram (optimal n was set to 4) language models on the cleaned GT set. In the testing phase, each word/bigram/trigram was evaluated in the language model. Sequences with low probabilities were marked as potential OCR errors and further processed in the candidate generation phase. For each word in the sequence we generated possible variants within a Levenshtein distance of 2 and checked the candidate lexicons for matches. We only signaled a positive match if a candidate was found.

LSTM Monochar - Team from ICTLab, USTH[3]*, Vietnam*

This approach relies on a Long-Short Term Memory (LSTM) model. The model was trained on the GT (lowercased, cleaned of special symbols) using a sliding window as an input and the last character as an expected output. The model then generates, from a given seed sequence, the probabilities of any character to follow. An OCR token is marked as erroneous if any character is associated with low probabilities regarding the LSTM prediction. Finally, a correction can be proposed by using the most probable character predicted by the model. To avoid any degeneration, the approach is currently limited to only one-character correction per token, which represent most of the errors anyway.

Seq2Seq - Team from GINI, Germany

This approach is based on the Seq2seq model [8] which has been specifically developed as a domain-independent model to match the input sequence to the output sequence. It is a six-layer recurrent net with LSTM cells [9] and attention mechanism [10] where the first two layers act as a decoder that matches the input character sequence of the OCR-ed text to a fixed-length vector, and the other four layers are an encoder that matches the vector to the output GT. The trained model corrects the erroneous tokens which enables us to identify the location of errors (by finding the difference between the input and output). TensorFlow has been used for model training.

BiLSTM - Team CMATER_JU from Jadavpur University, India

For the detection phase, words that do not belong to common dictionaries (found online and augmented with the given training data) are considered erroneous. The correction phase relies on a modified bi-directional LSTM. The training data is used to train a first RNN (RNN1) with two hidden LSTM layers. Similarly, a second RNN (RNN2) with two hidden LSTM layers is trained on reversed data (reversing word order in each sentence). For the correction of error words, 3 words or less to the left and right of the error word/words re selected and is input to the two RNNs with left input to RNN1 and right input to RNN2. Then the next words of each input are found. The top 4 words with Levenshtein distance to the error word lower than 3 are chosen. They are in turn input to the corresponding RNNs one by one and next words to all are found. All new words found in RNN1 are matched with 2. This outputs all phrases with a matched word, sorted by multiplying the probabilities obtained from the RNNs while generating the words.

MMDT [11] - Team from Institute for NLP, Univ Stuttgart, Germany

In order to account for the nature of errors that can occur in OCR text, our MMDT (Multi-Modular Domain-Tailored) approach combines a variety of modules for post-correction. The system proceeds in the two following stages: In a first stage, a set of specialized modules suggest corrected versions for the tokenized OCR text lines. Those modules can be context-independent (work on just one word at a time) or context-dependent (an entire text line is processed at a time). Specifically, we use statistical machine translation models on token and character-unigram levels, spell checking, compound word bigrams, split words into bigrams and introduce a module that learns text-specific vocabulary from the words appearing in the input text. The second stage is the decision phase. After

[2] Laboratoire de recherche en Informatique pluridisciplinaire

[3] University of Science and Technology of Hanoi

the collection of various suggestions per input token, these have to be ranked to enable a decision for the most probable output token given the context. We achieve this by assigning weights the different modules with the help of Minimal Error Rate Training (MERT).

Char-SMT/NMT - Team from CLUZH[4], Switzerland

Our method is based on ensembles of character-based Statistical and Neural Machine Translation (SMT/NMT) models trained exclusively on material released for the task. For each language and type (monographs, periodicals), we build several models on our internal training data (taking 10% as devset). The simplest models translate each token separately; context models translate each token within a window of 2 preceding and one succeeding tokens; factored Neural models additionally encode the time period (buckets of 50 years) as an additional feature. Neural models optionally include embeddings for glyphs. Our error detection algorithm uses the output of the MT systems: An error exists, if our model with the lowest Levenshtein distance on the devset proposes a change; or if the majority of the five best systems proposes a change; or if the OCR token did not occur in the corrected training set; or finally, if the token and one of its neighbors (translated/untranslated) do not occur in the training set, but their concatenation does. Our correction algorithm also relies on models with lowest Levenshtein distance on the devset. Suggestions of our best system result in exclusive candidates. Otherwise the translation frequency distribution is used as n-best list.

WFST-PostOCR - Team RAE-UAM from RAE[5], Spain

The proposed method is an application of the noisy channel model to the OCR error correction of English and French historical texts. Probabilistic character error models are estimated from the training corpus using longest common subsequence alignments of tokens and compiled into weighted finite-state edit transducers. With a maximum of one edition, the error model is applied to tokens, token splits, and concatenations of tokens, so that token segmentation errors can be addressed. Vocabulary and bigram language models are derived from the Google Books Ngram Corpus, which contains OCR errors but meets the characteristics of quantity and historical amplitude. Using the language model and the lattice of hypotheses generated by the error model, the best path is used to determine the best token sequence. Finally, since historical texts do not follow current standard spellings and typographical conventions, original token's case is applied to the system's output.

Anavec - Team TICCL from CLS[6], The Netherlands

Anavec[7] is a spelling correction system that stores words or n-grams from a lexicon and background corpus as anagram vectors, an unordered bag-of-characters model. Words to be corrected are similarly represented as anagram vectors and matched with the training data to find the closest neighbors. Anagram matches are resolved to actual correction candidates, which are in turn scored according to the vector distance, Levenshtein distance, frequency in background corpus, and presence in lexicon. Finally, a stack decoder algorithm (a beam search) incorporating also a language model component for context sensitivity, selects the most likely correction candidates given an input sequence. We trained this system with little to no regard for historical spelling to keep things simple; using Wikipedia as the background corpus and the lexicon from Aspell. Our system does both detection and correction, but we focused mainly on task 2. For task 1 we only participate for French as the English version has too many false positives as it is not aware of historical variants.

CLAM - Team from IITB[8]-Monash Research Academy, India

The proposed method relies on the Character Level Attention Model (CLAM) with beam search used at decoder's output. We used the open source system OpenNMT (http://opennmt.net) also used in [12]). Regarding the training phase: to take care of real word errors as well as non-word errors, at network's input we used the characters (with space as delimiter) from input OCR word o_t along with characters from few words (with \$ as delimiter) on its left ($o_{t-l:t-1}$) and right ($o_{t+1:t+r}$). At network's output we used the characters (with space as delimiter) from the Ground Truth word g_t corresponding the input OCR word o_t. $l = 4$,$r = 1$ gave the optimized results for first 3 datasets, and $l = 6$, $r = 1$ for last. Regarding the testing phase: we expect the model to jointly learn the language as well as error patterns in OCR output. Since our model avoids any changes on correct words, the unchanged one are considered correct in the detection phase, and the changed one are considered erroneous and are then used as suggestions for the correction phase. Our goal is to try out different contexts that are helpful to correct an OCR word and come up with the best context that give optimized F-Score and corrections.

2-pass-RNN - Team AMU-LIF-TALEP from LIF[9], France

The system used is a multilingual one. In order to detect possible errors in a raw OCR output, we propose an approach based on the use of RNNs. The full process is made in two passes, with relatively similar models. The first one is trained at character level and the second one at word level with features from the previous model aligned on each word, allowing the analysis to acknowledge errors at both character and word levels. For the correction part, we first use another neural network model to predict the error type of each character, if it is part of an erroneous sequence. We then detect the document language to compare each token to a specific dictionary and keep only the words which do not differ from the previously predicted sequence. This allows the number of possible answers to be reduced and we can thus compute a

[4] Institute of Computational Linguistics of the University of Zurich

[5] Centro de Estudios de la Real Academia Española

[6] Centre for Language Studies, Radboud University Nijmegen

[7] https://github.com/proycon/anavec

[8] Indian Institute of Technology Bombay

[9] Laboratoire d'Informatique Fondamentale de Marseille

LV distance on all candidates, which we weight with reversed POS-tagging probabilities and select the ones with the lowest weight. As a tie-breaker, we trained an n-gram language model to keep only the tokens with the highest probability.

EFP - Team from ICTLab/USTH[3]*, Vietnam*

This method[10] relies mostly on the Error Frequency Patterns (EFP) with some small variations. A rough filter was applied, in the sense that the method focuses only on tokens that are between 3 and 8 characters and ignores those including special symbols, punctuation and numbers. For Task 1, any tokens that do not belong to usual dictionaries are considered erroneous. For Task 2, the correction is done by using a pre-calculated table of error patterns. Candidates are generated by trying to switch each character (or pairs of characters) according to common error patterns, with the restriction the generated candidates belong to dictionaries.

IV. Results and discussion

Table II details the results for both Task 1 and 2 according to 4 subsets of the corpus: French language, English language, periodicals and monographs. The metrics (accuracy, recall, F-measure and %improvement) were calculated for each of these 4 subparts (technically considered as one aggregated document). The "E" symbol means the evaluation script ignored some of the proposed detection/corrections provided by the participants due to inconsistent offsets[11] (position not pointing to a token start). The "x" symbol corresponds to no exploitable results (e.g. incomplete participation or wrong non-fixable format). In the context of Task 2, the "-" symbol means that no global improvement was achieved. Some files may have been improved but the outcome is negative on average. The "=" symbol indicates an equal result for both the automatic and the semi-automatic approaches, which in most cases indicates that participants have provided only one candidate per correction.

The RAE-UAM team is the best performer on Task 1 with their WFST-PostOCR method achieving the best F-measure on every corpus subpart: from 0.55 on FR-monog and up to 0.73 on ENG-monog. Details on accuracies/recalls for each teams are given in Subsection IV-A.

The CLUZH team is the best performer on Task 2 with their Char-SMT/NMT method achieving the best improvement rate over every corpus subpart: from 29% on FR-period and up to 44% on the FR-monog. Global details on the number of documents improved for each team are given in Subsection IV-B.

Some participants have rightly pointed out some inaccuracies in the GT such as missing or incorrect corrections. The dataset, given its important size and its nature (manually annotated, OCR/GT automatically aligned) is obviously imperfect. Those inaccuracies, although rare, can still cause harm both during the training (by misleading the final model) and the evaluation phase (by wrongly considering a right correction). Major issues, such as inconsistent hyphenations (e.g. line breaks), were handled directly in the evaluation script, and as mentioned earlier related word corrections were therefore not taken into account for the evaluation.

A. Task 1: Error detection

Table III shows accuracy and recall scores on the error detection task. The best performing method appears to be WFST-PostOCR which also achieves the best recalls. Its accuracy is however overpassed by the 3 following other methods (MMDT, CLAM and Char SMT/NMT) but with a much lower recall. These indicators could be taken into account depending on the intended context of use of these techniques (recall is much more important if human validation is to follow, while precision seems more important to a fully automated process).

B. Task 2: Error correction

Table IV gives additional details on the number of documents involved and the quantity that each method was able to improve. For those who provided multiple correction candidates, we observe slightly better results for the automatic mode than for the semi-automatic mode, which shows the limited interest of this latest kind of evaluation.

V. Conclusion

This paper describes the first ICDAR competition on post-OCR text correction. The challenge consisted of two independent tasks: 1) error detection and 2) error correction on an original dataset of 12M OCR-ed symbols along with an aligned ground truth. The data come from sources of different natures (newspapers and monographs) and target 2 different languages. This competition demonstrated, through formatted results provided by the participants, the performance of their systems exposed to this specific dataset and to the custom metrics described in this paper.

Concerning the first task (error detection), the WFST-PostOCR method performs the best. Its authors proposed an application of the noisy channel model [13] where probabilistic character error models were estimated from the training corpus using longest common subsequence alignments of tokens and compiled into weighted finite-state edit transducers.

Concerning the second task (error correction), the Char SMT/NMT method performs the best with respect to the official metric (weighted sum of the Levenshtein distances). It is based on a set of character-based statistical and neural machine translation models trained exclusively on material released for the task.

In order to gather and share an overview of the approaches explored by the community, we invited the participants to submit their results even in case of low performances. Their feedback has shown that some of the participants were lacking time for training and testing their systems, while some others were struggling with formatting issues at the last minute. Thus, low performing methods should not necessary be assimilated to wrong research tracks. The submissions that resulted in low

[10] https://github.com/tung18tht/ICDAR-2017-Post-OCR-Correction

[11] To the benefit of participants who wrongly formated their results, manual shifts of 1 or -1 on the offsets were applied when observing a major improvement

TABLE II
SUMMARIZED RESULTS FOR TASKS 1 AND 2 ACCORDING TO CORPUS SPECIFICITY: FRENCH, ENGLISH, PERIODICALS AND MONOGRAPHS

	Task 1 (F-mesure)				Task 2 (%Improvement) Auto (top1) / Semi (weighted mean on top5)				
Corpus part >	ENG-mono.	ENG-period.	FR-mono.	FR-period.	ENG-mono.	ENG-period.	FR-mono.	FR-period.	
NbTokens (E.R.) >	63371 (10%)	33176 (15%)	32274 (5%)	48356 (7%)	63371 (10%)	33176 (15%)	32274 (5%)	48356 (7%)	
5gram-KN-LV	0.05	0.51	0.25	0.35	x	x	x	x	E
LSTM Monochar	x	x	0.17	x	- / -	x	- / -	x	
Seq2Seq	0.45	0.39	x	x	- / -	- / -	x	x	E
BiLSTM	0.09	0.06	0.05	0.05	- / -	x	x	x	E
2-pass-RNN	0.66	0.66	0.43	0.60	x	- / -	x	x	E
Anavec	x	x	0.24	0.42	5% / -	- / -	- / -	- / -	
WFST-PostOCR	**0.73**	**0.68**	**0.55**	**0.69**	28% / =	- / -	- / -	- / -	
CLAM	0.67	x	0.36	0.54	29% / =	22% / =	1% / =	5% / =	
Char-SMT/NMT	0.67	0.64	0.31	0.50	**43% / =**	**37% / =**	**44% / =**	**29% / =**	
EFP	0.69	0.54	0.40	0.54	13% / 11%	- / -	23% / =	5% / 4%	
MMDT	0.66	0.44	0.36	0.41	20% / =	- / -	3% / =	2% / =	E

TABLE III
COMPARED PERFORMANCES (ACCURACY / RECALL) FOR TASK 1

Corpus part	ENG-mono.	ENG-period.	FR-mono.	FR-period.
Nb tokens	63371	33176	32274	48356
(Err.Rate.)	(10%)	(15%)	(5%)	(7%)
5gram-KN-LV	0.20 / 0.03	0.50 / 0.53	0.17 / 0.46	0.26 / 0.52
LSTM Monochar	x	x	0.26 / 0.12	x
Seq2Seq	0.36 / 0.59	0.35 / 0.44	x	x
BiLSTM	0.21 / 0.06	0.25 / 0.03	0.06 / 0.05	0.09 / 0.04
2-pass-RNN	0.58 / 0.77	0.64 / 0.68	0.33 / 0.60	0.54 / 0.67
Anavec	x	x	0.18 / 0.37	0.40 / 0.43
WFST-PostOCR	**0.67 / 0.82**	**0.68 / 0.68**	**0.51 / 0.59**	**0.72 / 0.66**
CLAM	0.93 / 0.52	x	0.48 / 0.28	0.71 / 0.44
Char-SMT/NMT	0.98 / 0.51	0.88 / 0.50	0.74 / 0.19	0.93 / 0.34
EFP	0.62 / 0.77	0.54 / 0.55	0.29 / 0.60	0.49 / 0.58
MMDT	0.84 / 0.55	0.72 / 0.32	0.62 / 0.25	0.71 / 0.28

TABLE IV
NUMBER OF DOCUMENTS GLOBALLY IMPROVED WITH TASK 2 FOR BOTH MODES: (AUTO-/SEMI-AUTOMATIC)

Corpus part	ENG-mono.	ENG-period.	FR-mono.	FR-period.
Nb docu.	41	4	54	12
LSTM Monochar	7 / 7	x	2 / 4	x
Seq2Seq	0 / 0	0 / 0	x	x
BiLSTM	0 / 0	x	x	x
2-pass-RNN	x	0 / 0	x	x
Anavec	7 / 0	0 / 0	12 / 3	2 / 0
WFST-PostOCR	36 / 36	0 / 0	20 / 20	3 / 3
CLAM	36 / 36	4 / 4	31 / 31	7 / 7
Char-SMT/NMT	**40 / 40**	**4 / 4**	**46 / 46**	**12 / 12**
EFP	31 / 33	1 / 0	24 / 25	11 / 10
MMDT	37 / 37	3 / 3	28 / 28	7 / 7

scores in the context of this competition could of course work better on different conditions (datasets, languages, formats and metrics).

In perspective, it would be interesting to test a less complex format for the evaluation with full sequences provided as an input instead of a list of positions/corrections. This would require a posterior automatic alignment phase (e.g. [14]) with its pros (easier for participants), its cons (difficult support of multiple correction candidates) and its risks (miss-alignment).

In a nutshell, this competition has illustrated the difficulty of the proposed tasks as only half of the submitted approaches succeeded in enhancing the existing OCR. However, this competition also highlights the strong interest of the community for this topic, which is of primary interest for enhancing the access to patrimonial content from digital libraries.

ACKNOWLEDGMENTS

We would like to thanks all the participants for their involvement. Without them, the competition would have not happened. This competition was co-organized by the L3i Laboratory and the National Library of France (BnF) and was supported by the 8^{th} BnF quadrennial research plan 2016-19.

REFERENCES

[1] G. Chiron, A. Doucet, M. Coustaty, J.-P. Moreux, and M. Visani, “Impact of ocr errors on the use of digital libraries - towards a better access to information,” in *JCDL’17, ACM/IEEE-CS Joint Conference on Digital Libraries, June 2017, Toronto, Ontario, Canada*, 2017/06 2017.

[2] M. C. Traub and al., “Impact analysis of ocr quality on research tasks in digital archives,” in *International Conference on Theory and Practice of Digital Libraries*. Springer, 2015, pp. 252–263.

[3] O. Kolak and P. Resnik, “Ocr error correction using a noisy channel model,” in *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 2002, pp. 257–262.

[4] 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*. ACM, 2014, pp. 45–51.

[5] M. Reynaert, “Ticclops: Text-induced corpus clean-up as online processing system.” in *COLING (Demos)*, 2014, pp. 52–56.

[6] Y. Bassil and M. Alwani, “Ocr post-processing error correction algorithm using google’s online spelling suggestion,” *Journal of Emerging Trends in Comp. and Info. Sciences*, vol. 3, 2012.

[7] A. Abdulkader and M. R. Casey, “Low cost correction of ocr errors using learning in a multi-engine environment,” in *Document Analysis and Recognition, 2009. ICDAR’09. 10th International Conference on*. IEEE, 2009, pp. 576–580.

[8] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[10] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *preprint arXiv:1409.0473*, 2014.

[11] S. Sarah and K. Jonas, “Multi-modular domain-tailored ocr post-correction,” in *Empirical Methods in Natural Language Processing (EMNLP), 2017 Conference on*, 2017.

[12] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” *preprint arXiv:1508.04025*, 2015.

[13] E. Brill and R. C. Moore, “An improved error model for noisy channel spelling correction,” in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000, pp. 286–293.

[14] I. Z. Yalniz and R. Manmatha, “A fast alignment scheme for automatic ocr evaluation of books,” in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 754–758.