2014 14th International Conference on Frontiers in Handwriting Recognition

Automatic Line Segmentation and Ground-Truth Alignment of Handwritten Documents

Théodore Bluche*†, Bastien Moysset*, and Christopher Kermorvant*
*A2iA SA, Paris, France
†LIMSI CNRS, Spoken Language Processing Group, Orsay, France

***Abstract*—In this paper, we present a method for the automatic segmentation and transcript alignment of documents, for which we only have the transcript at the document level. We consider several line segmentation hypotheses, and recognition hypotheses for each segmented line. The recognition is highly constrained with the document transcript. We formalize the problem in a weighted finite-state transducer framework. We evaluate how the constraints help achieve a reasonable result. In particular, we assess the performance of the system both in terms of segmentation quality and transcript mapping. The main contribution of this paper is that we jointly find the best segmentation and transcript mapping that allow to align the image with the whole ground-truth text. The evaluation is carried out on fully annotated public databases. Furthermore, we retrieved training material with this system for the Maurdor evaluation, where the data was only annotated at the paragraph level. With the automatically segmented and annotated lines, we record a relative improvement in Word Error Rate of 35.6%.**

I. Introduction

To train automatic text recognition systems, we need annotated lines of text, which are time consuming – hence expensive – to obtain through manual segmentation and annotation. Public databases of images of handwritten text, annotated at the line level are rare. Some examples include the Rimes [1] and IAM [2] databases. On the other hand, many databases of documents are available. We can easily retrieve both the images and the transcript at the document level. Finally, we observe that in the construction of public databases, people are often asked to copy a text (*e.g.* from the LOB corpus for IAM, from newswire and the web for OpenHaRT). In such cases, the transcript is already known because defined prior to the actual handwriting action.

For these reasons, *i.e.* to relax the annotation effort, and benefit from the wealth of transcribed documents available on the web, general methods for retrieving the line segmentation and annotation would be helpful. Numerous line segmentation algorithms exist, all having some strengths and weaknesses. The document transcript, on the other hand, is a reliable source of information, that puts many constraints on what the lines content is expected to be. Including those constraints in a handwritten text recognizer could help to choose the most suited line segmentation, and to map the transcript to the segmented lines.

This problem has already been addressed in the literature (Section II). Proposed methods are either recognition-based or not. In most papers, the line positions are assumed to be either known or reliably retrieved. In this paper, we propose to improve those methods. First, we take into account several line segmentation hypotheses, rather than the result of a single segmenter. Then, we include several constraints, at different level: line ordering, limited search space for the transcript mapping using a text recognizer, word ordering in the transcript. We formalize the problem in a weighted Finite-State Transducer (FST) framework. Each step – segmentation, recognition, transcript order – is represented as an FST. The constraints are combined by composition of these FSTs [3], and the shortest path in the composed FST corresponds to a line segmentation and a transcript mapping. To the best of our knowledge, this is the first attempt at jointly finding a good line segmentation and a good mapping. The proposed method has several limitations (it cannot cope with transcript errors or complex page layouts), but we report promising results for simple tasks and for the Maurdor evaluation [4].

The paper is divided as follows. Section II contains a brief literature review of related systems. The proposed method is described in Section III. Then, we present the experimental setup in Section IV. The results of the evaluation of the method are summarized in Section V. We propose in Section VI some perspectives for further improvements, before concluding in Section VII.

II. Relation to prior work

The problem of mapping transcript to images has motivated some research in the past decade, either for the alignment of Optical Character Recognition (OCR) output with book content (e.g. in [5], [6]), or for mapping the transcription of historical documents to segmented words or lines (such as [7], [8]).

Feng et al. propose in [5] a method for aligning OCR output with the entire book using Hidden Markov Models (HMMs). They first align anchor words, which appear only once in the OCR output, then words between anchors, and finally characters between matched words. A similar method using anchor words is presented in [9]. It is a recognition-based method which computes a distance between a word in a limited lexicon and a word image with dynamic programming. A post-processing step is applied to recover from segmentation errors. Kornfield et al. [8] argue that handwriting recognition systems are not good enough to help the mapping of text to historical document images. They match words of the transcript with automatically segmented word images using Dynamic Time Warping (DTW).

Other methods are recognition-based. Rothfeder et al. [7] automatically segment the images into words, and compute word-level features. Then, they use a linear HMM which

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.117

can cope with segmentation errors to align images with the transcript. Fischer et al. [10] focus on the alignment of inaccurate transcriptions using an HMM recognition system. They extract features from line images, and use a lexicon containing only the words of the page, and a bigram language model. In [11], experiments are conducted on the St. Gall database, where the transcription is known, but the line breaks are not indicated. They perform Viterbi alignment with HMMs corresponding to the transcript, including spelling variants, and model replacement for unknown characters. In [12], the authors use an FST framework for the ground-truth alignment in difficult historical documents. They represent the transcript as an FST with variants, allowing the system to cope with OCR errors, ligatures, and hyphenation. They extract OCR lattices, and align them using several approaches.

These methods assume a good line segmentation, although this problem is not trivial [13]. In our work, we kept several aspects which seemed interesting, namely the recognition-based approach, the FST framework, and the concept of forced alignments. Our contributions lie in the different levels of constraints we add, particularly in the segmentation alternatives, which allow us to rely less on the line segmentation algorithm. The final segmentation and mapping are jointly found.

III. Description of the Method

A. Overview

The proposed method takes two inputs: the document image(s) and transcript. The goal of the method is to retrieve (i) a *correct* line segmentation – *i.e.* all parts of the image corresponding to the considered transcript should be retrieved –, and (ii) a good mapping – *i.e.* all words of the transcript should be assigned to the corresponding line images.

The method consists of several successive steps. First, we generate line segmentation hypotheses from the document (or paragraph) images. Then, each line hypothesis is passed through a recognizer to generate hypotheses of word sequences. This recognition phase can be highly constrained by the transcript, *e.g.* in terms of word orderings. Then, segmentation and recognition hypotheses are combined, and more transcript constraints are added, in order to get a consistent mapping of transcript words to line images. Finally, we select the best hypotheses in that constrained set.

We formulate the problem in a weighted FST framework. This allows to model the sequential aspect of the task, to represent easily the different hypotheses, and to encode the constraints and combine them by means of FST composition [3]. In the following, we present how each step can be encoded as an FST, and how the FSTs we obtain are composed to enrich the hypotheses and representations, and add the constraints. The FSTs are illustrated on Fig. 1.

B. Segmentation

The main risk of the segmentation step is to miss relevant lines in the image. If this happens, we could not map the corresponding words of the transcript to that image area. Hence, we should favor over segmentation, possibly with perfect recall, even if the precision is low. The later stages of the approach, in particular the constraints, will ensure that irrelevant lines are ignored in the final segmentation.

Fig. 2 shows that each segmentation algorithm yields a different result. Each algorithm has its strengths and weaknesses, or can be more suited to some kinds of documents than others. For illustration, an evaluation of several line segmentation algorithms with different metrics has been carried out in [13]. We want to take advantage of the strengths of the different algorithms. Since irrelevant lines should be ignored with the constraints, and we want a very high recall, we applied several segmentation algorithms.

Fig. 2. Different segmentation algorithms yield different results.

The projection-based algorithm computes a smoothed horizontal projection profile. From the profile, we obtain vertical line boundaries using different thresholds, in a watershed-like fashion. The horizontal boundaries are calculated by removal of white pixels on both sides. This method gives a hierarchical segmentation with several hypotheses. The shredding algorithm is explained in [14]. Finally, a rectangle-based filtering algorithm is applied, which uses a median filtering with a rectangular mask. This technique is inspired from [15], and explained in [13]. This segmentation procedure returns rectangular bounding boxes for line hypotheses.

The first constraint we can implement concerns the line ordering. Indeed, we can assume that the transcript order corresponds to a natural reading order, which, in the image, would be top-to-bottom, left-to-right (for Latin-script languages). Therefore, we order the line hypotheses accordingly, so a line at the bottom of the document cannot be aligned with the beginning of the transcript in subsequent steps.

The line ordering is represented as a directed acyclic graph. First the lines are ordered by ascending order of their vertical position (*i.e.* the coordinate of the top of the bounding box). Then, we consider each line in turn, and create a link with a previous line if the vertical overlap between the boxes is not bigger than half the height of the smallest, and if there is no other existing path in the graph between the considered lines. Lines without predecessor correspond to possible start nodes, and lines without successor are final nodes. The segmentation FST (S) is obtained by labeling the arcs with line identifiers (as input and output symbol). We leave the possibility to skip irrelevant lines by adding a parallel arc to every line arc with the line identifier input and a special output symbol `#skip`.

A skip transition does not have any cost, and the segmentation arcs have a cost σ (which we can see as a line insertion penalty). This scoring scheme could be improved if the segmentation algorithms also returned a confidence score for each line. Note that the graph creation procedure should be improved to handle images with several columns, side or margin notes, or complicated layouts in general.

Fig. 1. Segmentation S, Recognition R and Transcript T FSTs.

C. Recognition

We run the recognizer on each segmented line individually. We apply the preprocessing and feature extraction used to train the recognizer (see Section IV-C and [16]), and we run the decoder with lexical and grammatical constraints (also represented as FSTs) to extract FST lattices [17].

The lexicon is limited to the words present in the transcript. The grammar is the transcript with skips. It is a linear acceptor corresponding to the transcript with all states being initial and final states. A valid path is a substring of the transcript. When we know the line breaks in the transcript, we can put all line transcripts in parallel and force the recognizer to output full lines of transcript hypotheses (only the states at line breaks positions can be initial/final). To some extent, we can view the recognition step as a sort of forced alignment.

The recognition FST (R) is the union of the lattice FSTs. We modify the lattice FSTs so that output symbols are recognized words and input symbols are the line identifiers. The weights are the unscaled optical model scores. We connect all lattices start and final states to a shared looping start/end state. We add a loop to this state with a `#skip:#skip` transition and skip cost to allow line hypothesis rejection. The composed $S \circ R$ FST reads lines and outputs words, and valid paths correspond to a segmentation and a mapping of transcript subsequences.

D. Transcript constraint

The transcript FST (T) is a linear acceptor corresponding to the sequence of words in the transcript. Moreover, to each state corresponds a state that absorbs skips: when reading a skip at position i, move to skip-state i. We can stay in this state while we keep reading skips, continue in the transcript if we read word $i+1$, or transit to the state for the k-th word in the transcript, if we read word $i+k$. In the last case, we can add a penalty depending on the value of k, to prevent missing alignment to happen too often. In the limit case of an infinite penalty, the system is forced to map the whole transcript.

E. Finding the best mapping

When we compose $S \circ R$ with T, we get an FST in which inputs are lines and the output is the transcript. So a valid path in this transducer corresponds to a segmentation (when looking at the input sequence) and to a transcript mapping, respecting the transcript order (when looking at the output sequence). The shortest path in $S \circ R \circ T$ should correspond to the best segmentation and annotation of the image given the document transcript.

IV. Experimental Setup

A. Evaluation methodology

The previous sections described the general ideas for jointly finding the best segmentation and mapping of transcript using recognition. Visual inspection of the results is a good way to check that we achieve something, but not sufficient, especially for big databases, to assess the importance of specific constraints or system components.

We try to solve two problems: annotation and segmentation. Both are evaluated with well-known metrics, for databases where ground-truth for line position and annotation is available. To evaluate the segmentation, we used the ZoneMap metric, developed by the Laboratoire Nationnal de métrologie et d'Éssais (LNE). The metric first tries to map line hypotheses to reference positions (this is a sort of bounding box alignment). It takes into account all possible configurations: matches, misses, false alarms, splits and merges (in the result tables: Ma, Mi, FA, S, Me), and a measure of error is calculated.

For the transcript mapping, we are interested in two aspects. First, we want the whole transcript to be mapped to the image. Thus, we can compare the reference transcript with the obtained mapping. We use the Levenstein algorithm to align the actual transcript with the mapped one, and record the edits (corrects, substitutions, deletions, insertions – in result tables: C, S, D, I) achieving the minimum edit distance, and compute a Word Error Rate (WER)[1]. We want the segmented lines to hold the correct transcript. We also perform this alignment at the line level, using the bounding box alignments from ZoneMap.

B. Databases

To assess the quality, the advantages and weaknesses of this system, we will use databases for which we know the ground-truth for both the line segmentation and the line transcription. Such databases are publicly available and extensively used in automatic text recognition problems.

The Rimes database [1] consists of a training set of 1,500 images of handwritten paragraphs in French, and an evaluation set of 100 images. The IAM database [2] consists of 747 images of handwritten documents in English for training, 116 for validation, and 336 for evaluation. Examples of image are shown on Fig. 3. We carried out the experiments on parts of the databases which were not used to train the recognizers.

C. Recognition systems

The recognition systems we used in our experiments, unless stated otherwise, are HMMs with Gaussian Mixture Models

[1]Note that this is **not** a recognition error, but a mapping error

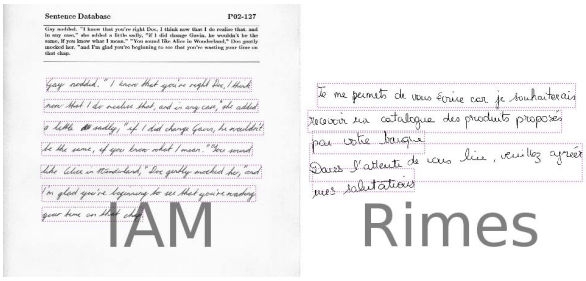

Fig. 3. Examples of image from IAM (left) and Rimes (right) databases. Dotted lines represent the ground-truth segmentation.

(GMMs), trained on the training set of the considered database. In some experiments, other recognizers are plugged into the system. They are either GMM-HMMs, or hybrid Bidirectional Long Short-Term Memory Recurrent Neural Networks (BLSTM-RNN) / HMM. The image pre-processing, feature extraction, system architectures and training procedures are explained in [16], for Rimes and IAM. The GMM-HMM trained on (a subset of French handwritten) Maurdor data is the same as the one for Rimes, except for the training material. The data (text and images) used for this evaluation were not seen during the training of the recognition systems.

V. Evaluation and Results

A. How keeping several segmentation alternatives helps?

In this section, we study the choice of segmentation, and how the segmentation and transcript mapping relate to each other. We compare different segmentation algorithms in isolation, the ground-truth segmentation, and keeping all segmentation alternatives. The results are presented on Table I. For each segmentation method, we report the segmentation results without and with transcript mapping, to study the effect of transcript mapping on segmentation. We observe that the mapping constraint generally improves ZoneMap error on IAM, while on Rimes the error increases in most cases. The mapping also decreases the number of splits and false alarms, as expected, at the expense of an increase of merges and misses.

Actually, the transcript constraints allow to skip lines to deal with over-segmentation. Skipping too many lines results in an increase of misses, which in turn increases the document-level mapping deletions, hence the WER. When we keep several segmentation hypotheses, we reduce the number of misses, which may explain why we get better results than the ground-truth segmentation (in terms of mapping) for Rimes. Note also that when we only want to create training material, missing a few lines is not a big issue, and the interesting part is to get the good transcript for retrieved lines (then an appropriate measure would be the line-level mapping WER minus the document-level deletions).

When we keep all segmentation hypotheses, the segmentation and mapping errors are not as good as when we consider only the best segmenter, but they are close. Since we usually do not know *a priori* which segmenter will be best on new data, keeping all hypotheses seems to be a good compromise. Table II shows a correlation between segmenter performances and lines kept in the final result on IAM, which is not observed on Rimes.

TABLE II. Line hypotheses per segmenter before and after mapping. The errors correspond to the segmentation and mapping using each algorithm in isolation.

		Shredding	Rectangle	Profile
IAM	Segm. Err.	0.77	6.03	0.87
	Line WER	1.24	4.48	0.97
	All lines	1,096 (35.9%)	984 (32.2%)	977 (32.0%)
	Final lines	642 (62.76%)	93 (9.09%)	288 (28.15%)
Rimes	Segm. Err.	12.57	5.37	9.51
	Line WER	11.07	2.73	4.45
	All lines	805 (29.29%)	777 (28.28%)	1166 (42.43%)
	Final lines	323 (39.58%)	181 (22.18%)	977 (31.96%)

Moreover, we studied the effect of knowing the line breaks in the transcript. We see that the segmentation results are only very slightly improved. However, the transcript mapping results are much better for IAM, but not for Rimes, where line misses and merges increase.

B. The influence of the constraints

We presented constraints that we argued should help solve this problem. In Table III, we show the results of relaxing two constraints. In the first experiment, we removed the composition of the recognition FST with the transcript FST (we search in $S \circ R$, not in $S \circ R \circ T$). Thus, mapped transcripts may overlap accross lines (see the insertions and substitutions in the document-level WER), and may not be in the right order. For the segmentation, T plays a role in the decrease of splits.

In the second experiment, we removed the grammar in recognition, *i.e.* we can only recognize words of the transcript but in any order. The order is still implemented by the transcript FST. We observe a lot of misses, resulting in deletions of transcript words, probably because the lattices were not rich enough to include correct word sequences, hence the corresponding parts were skipped in the annotation. Thus the transcript order constraint, which implements a semi-forced alignment, is important for the mapping to take place efficiently.

C. Do we need a very good recognition system?

In the previous experiments, we used a GMM-HMM trained on the training set of the same database. We can expect [16] that the recognizer is good for the new images, which should have approximately the same distribution as the training images. For practical usage of this system, we may not have access to a training set. To simulate this scenario, we plugged different GMM-HMMs, trained on different databases. The results are presented on Table IV. For IAM, the GMM-HMMs trained on French data (Rimes and Maurdor) are not as good as the in-domain recognizer, yet the error rates are in reasonable ranges. For Rimes, the GMM-HMM trained on Maurdor data seems even slightly better, while the recognizers trained on IAM have a high level of misses/deletions, due to the absence of modeling of accentuated characters. We also applied BLSTM-RNNs, which outperform GMM-HMMs in handwriting recognition (cf. [16], *e.g.*). Interestingly, for IAM, the RNN trained on Rimes yields very close results to the one trained on IAM.

TABLE I. EFFECT OF THE SEGMENTATION AND LINE BREAK SYMBOLS

IAM	Segmentation						Document level mapping					Line level mapping				
	Ma	Me	S	Mi	FA	Err.	C	S	D	I	WER	C	S	D	I	WER
Ground-Truth lines + mapping	976	0	0	0	0	0.00	100.00	0.00	0.00	0.00	0.00	99.76	0.00	0.24	0.24	0.47
Shredding segmentation	874	0	102	0	0	1.56	-	-	-	-	-	-	-	-	-	-
+ mapping	965	3	5	0	0	0.77	99.60	0.00	0.40	0.00	0.40	99.18	0.00	0.82	0.42	1.24
Rectangle segmentation	953	0	15	8	0	4.90	-	-	-	-	-	-	-	-	-	-
+ mapping	949	0	0	24	0	6.03	97.27	0.00	2.73	0.00	2.73	96.40	0.00	3.60	0.88	4.48
Profile segmentation	972	0	2	2	0	1.56	-	-	-	-	-	-	-	-	-	-
+ mapping	971	1	0	2	0	0.87	99.75	0.00	0.25	0.00	0.25	99.39	0.00	0.61	0.36	0.97
All segmentations	2	0	974	0	0	282.38	-	-	-	-	-	-	-	-	-	-
+ mapping	947	1	26	0	0	0.90	99.75	0.00	0.25	0.00	0.25	99.26	0.01	0.73	0.48	1.22
Use line breaks	972	0	3	1	0	0.82	99.82	0.00	0.18	0.00	0.18	99.80	0.00	0.20	0.02	0.22
RIMES	**Ma**	**Me**	**S**	**Mi**	**FA**	**Err.**	**C**	**S**	**D**	**I**	**WER**	**C**	**S**	**D**	**I**	**WER**
Ground-Truth lines + mapping	739	16	0	7	0	4.22	97.00	0.00	3.00	0.00	3.00	96.05	0.39	3.56	0.57	4.52
Shredding segmentation	739	4	30	1	1	6.04	-	-	-	-	-	-	-	-	-	-
+ mapping	645	56	0	20	1	12.57	91.15	0.00	8.85	0.00	8.85	89.71	0.66	9.63	0.78	11.07
Rectangle segmentation	771	2	2	1	0	3.68	-	-	-	-	-	-	-	-	-	-
+ mapping	757	6	0	9	0	5.37	98.90	0.00	1.10	0.00	1.10	97.89	0.39	1.72	0.62	2.73
Profile segmentation	574	5	194	0	2	82.03	-	-	-	-	-	-	-	-	-	-
+ mapping	724	16	11	11	1	9.51	97.82	0.00	2.18	0.00	2.18	96.49	0.39	3.12	0.94	4.45
All segmentations	1	0	777	0	3	344.57	-	-	-	-	-	-	-	-	-	-
+ mapping	744	10	10	4	2	7.40	98.95	0.00	1.05	0.00	1.05	97.82	0.39	1.79	0.74	2.93
Use line breaks	723	16	10	13	0	9.06	98.44	0.00	1.56	0.00	1.56	97.62	0.37	2.00	0.44	2.82

TABLE III. CONTRIBUTION OF GENERAL CONSTRAINTS

IAM	Segmentation						Document level mapping					Line level mapping				
	Ma	Me	S	Mi	FA	Err.	C	S	D	I	WER	C	S	D	I	WER
All constraints	947	1	26	0	0	0.90	99.75	0.00	0.25	0.00	0.25	99.26	0.01	0.73	0.48	1.22
No transcript FST	929	0	46	1	0	0.75	99.55	0.20	0.25	2.76	3.21	99.52	0.20	0.28	2.80	3.28
No order in recognition	171	13	8	771	0	88.85	10.94	0.00	89.06	0.00	89.06	10.01	0.69	89.30	0.25	90.24
RIMES	**Ma**	**Me**	**S**	**Mi**	**FA**	**Err.**	**C**	**S**	**D**	**I**	**WER**	**C**	**S**	**D**	**I**	**WER**
All constraints	744	10	10	4	2	7.40	98.95	0.00	1.05	0.00	1.05	97.82	0.39	1.79	0.74	2.93
No transcript FST	704	2	69	1	1	7.95	97.39	1.01	1.60	4.03	6.63	96.49	1.33	2.18	4.61	8.12
No order in recognition	413	65	2	227	1	44.94	52.35	0.00	47.65	0.00	47.65	51.14	0.53	48.32	0.67	49.53

TABLE IV. INFLUENCE OF THE CHOICE OF RECOGNITION SYSTEM

IAM	Segmentation						Document level mapping					Line level mapping				
	Ma	Me	S	Mi	FA	Err.	C	S	D	I	WER	C	S	D	I	WER
GMM-HMM IAM	947	1	26	0	0	0.90	99.75	0.00	0.25	0.00	0.25	99.26	0.01	0.73	0.48	1.22
GMM-HMM Rimes	929	0	46	1	0	1.32	99.83	0.00	0.17	0.00	0.17	98.74	0.01	1.25	1.08	2.34
GMM-HMM Maurdor	934	0	42	0	0	0.84	99.87	0.00	0.13	0.00	0.13	98.83	0.03	1.13	1.00	2.17
BLSTM-RNN IAM	973	0	3	0	0	0.80	100.00	0.00	0.00	0.00	0.00	99.94	0.00	0.06	0.06	0.11
BLSTM-RNN Rimes	972	0	4	0	0	1.06	100.00	0.00	0.00	0.00	0.00	99.92	0.00	0.08	0.08	0.16
RIMES	**Ma**	**Me**	**S**	**Mi**	**FA**	**Err.**	**C**	**S**	**D**	**I**	**WER**	**C**	**S**	**D**	**I**	**WER**
GMM-HMM IAM	726	6	16	24	1	8.81	96.45	0.00	3.55	0.00	3.55	95.30	0.39	4.31	0.76	5.46
GMM-HMM Rimes	744	10	10	4	2	7.40	98.95	0.00	1.05	0.00	1.05	97.82	0.39	1.79	0.74	2.93
GMM-HMM Maurdor	740	7	22	2	2	6.38	99.50	0.00	0.50	0.00	0.50	98.28	0.39	1.33	0.83	2.55
BLSTM-RNN IAM	736	4	17	17	1	7.16	97.36	0.00	2.64	0.00	2.64	96.54	0.39	3.07	0.43	3.88
BLSTM-RNN Rimes	752	5	15	1	2	6.11	99.72	0.00	0.28	0.00	0.28	98.85	0.39	0.76	0.48	1.63

The constraints we introduced seem appropriate. Indeed, even with a recognition system trained on a different distribution of data achieves good results. It is however important for the recognizer to model the characters of the new transcripts. Moreover, a powerful recognition system, such as a BLSTM-RNN, even if trained on different language and images, improves the performance, which indicates that the recognizer quality matters.

D. A practical usage: getting training material for the Maurdor evaluation

In this section, we present a typical use case for this kind of system. For the Maurdor evaluation, the training data were segmented and annotated into text zones, which did not always correspond to single lines of text. For the English data, there were 7,357 zones with only one line, and 940 zones containing several lines. The positions of these lines are not given, but the transcript of the zone contains line break symbols. From these, we calculated a total number of lines in the training set of 11,608. We cannot evaluate the results as in previous sections. Instead, we trained a Multi-Dimensional (MD)LSTM-RNN with the available training material, and computed the WER of the complete recognition system on the test set. The recognition system, including the RNN, lexicon and language model details, and the line segmentation and decoding process are thoroughly presented in [4]. The results are presented in Table V.

First, we trained an MDLSTM-RNN on single-line zones.

TABLE V. EVOLUTION OF RNN PERFORMANCE AFTER EACH LOOP OF AUTOMATIC DATA ANNOTATION.

RNN training material	# lines / % of max.	WER
Initialization: only single-line zones	7,310 / 63.0	54.7%
First pass of transcript mapping	10,570 / 91.1	43.8%
Second pass of transcript mapping	10,925 / 94.1	35.2%

This model is used to segment and map transcript of multi-line zones. With this method, we retrieved 3,260 new lines for training. That is a relative increase of 45%. We trained a new MDLSTM-RNN with the new training set, and the WER drops from 54.7% to 43.8% (19.9% relative WER improvement). We perform a new transcript mapping with this RNN. We only retrieve 355 more lines, but the training material is better. Indeed, if we train a third MDLSTM-RNN with that data, we record another 19.2% relative WER improvement on the test set. This method played a key role in our success in the evaluation [4].

VI. PERSPECTIVES

The different aspects of the method could be improved. For the line segmentation FST, a sensible way of assigning different weights to the hypotheses would be beneficial, e.g. if the algorithms returned a confidence score. We should also build the segmentation graph so that complex layouts are properly handled. The scoring scheme for the recognition (*i.e.* replacing the mere recognizer scores) could also be improved for a better line rejection. Moreover, our method cannot cope with transcription errors, and it could be implemented. Finally, we plan to evaluate our method on difficult historical documents, to compare the results with other publications.

VII. CONCLUSION

We presented a method for the automatic line segmentation and annotation of documents, when only the image and whole transcript are available. While taking inspiration from previous published works, we introduced new constraints. We evaluated the method on public databases of handwritten text, where the ground-truth for line positions and transcript is known. Both the segmentation and the mapping are evaluated with standard metrics, and the importance of the different aspects of our system are proved: the ordered multiple segmentation hypotheses, the transcript contraints on word ordering at line and document level, the constrained recognition with different recognizers. The success of the approach is strengthened by its application in the Maurdor evaluation, where we had to retrieve the training material from images of multi-line zones. The performance was underlined by the recognition results of a MDLSTM-RNN trained with the new material, which were improved by 35.6% compared with a system using only single-line zones. This method allowed us to train the systems that won the Maurdor evaluation.

ACKNOWLEDGEMENT

This work was partially funded by the French Defense Agency (DGA) through the Maurdor research contract with Airbus Defence and Space (Cassidian), and by the French Grand Emprunt-Investissements d'Avenir program through the PACTE project.

REFERENCES

[1] E. Augustin, M. Carré, E. Grosicki, J.-M. Brodin, E. Geoffrois, and F. Preteux, "RIMES evaluation campaign for handwritten mail processing," in *Proceedings of the Workshop on Frontiers in Handwriting Recognition*, no. 1, 2006.

[2] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, Nov. 2002.

[3] M. Mohri, "Weighted Finite-State Transducer Algorithms. An Overview," *Studies In Fuzziness And Soft Computing*, vol. 148, pp. 1–13, 2004.

[4] B. Moysset, T. Bluche, M. Knibbe, M.-F. Benzeghiba, R. Messina, J. Louradour, and C. Kermorvant, "The A2iA Multi-lingual Text Recognition System at the Maurdor Evaluation," in *14th International Conference on Frontiers in Handwriting Recognition (ICFHR-2014)*, 2014.

[5] S. Feng and R. Manmatha, "A hierarchical, HMM-based automatic evaluation of OCR accuracy for a digital library of books," *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries - JCDL '06*, pp. 109–118, 2006.

[6] 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.

[7] J. Rothfeder, R. Manmatha, and T. M. Rath, "Aligning Transcripts to Automatically Segmented Handwritten Manuscripts," in *Proc. 7th Int. Workshop on Document Analysis Systems*, 2006, pp. 84–95.

[8] E. Kornfield, R. Manmatha, and J. Allan, "Text alignment with handwritten documents," in *First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings.* IEEE, 2004, pp. 195–209.

[9] C. Huang and S. N. Srihari, "Mapping Transcripts to Handwritten Text," in *Tenth International Workshop on Frontiers in Handwriting Recognition.*, 2006.

[10] A. Fischer, E. Indermuhle, V. Frinken, and H. Bunke, "HMM-Based Alignment of Inaccurate Transcriptions for Historical Documents," in *2011 International Conference on Document Analysis and Recognition*. Ieee, Sep. 2011, pp. 53–57.

[11] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, "Transcription alignment of Latin manuscripts using hidden Markov models," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing - HIP '11*. New York, New York, USA: ACM Press, Sep. 2011, p. 29.

[12] M. Al Azawi, M. Liwicki, and T. M. Breuel, "WFST-based ground truth alignment for difficult historical documents with text modification and layout variations," in *Document Recognition and Retrieval*, R. Zanibbi and B. Coüasnon, Eds., Feb. 2013, pp. 865 818–865 818–12.

[13] B. Moysset and C. Kermorvant, "On the Evaluation of Handwritten Text Line Detection Algorithms," *2013 12th International Conference on Document Analysis and Recognition*, pp. 185–189, Aug. 2013.

[14] A. Nicolaou and B. Gatos, "Handwritten Text Line Segmentation by Shredding Text into its Lines," *International Conference on Document Analysis and Recognition*, pp. 626–630, 2009.

[15] A. Zahour, L. Likforman-Sulem, W. Boussellaa, and B. Taconet, "Text Line Segmentation of Historical Arabic Documents," in *International Conference on Document Analysis and Recognition*, no. Icdar. IEEE, Sep. 2007, pp. 138–142.

[16] T. Bluche, H. Ney, and C. Kermorvant, "A Comparison of Sequence-Trained Deep Neural Networks and Recurrent Neural Networks Optical Modeling for Handwriting Recognition," in *2nd International Conference on Statistical Language and Speech Processing (SLSP2014)*, 2014 - Submitted.

[17] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlicek, Y. Qian *et al.*, "Generating exact lattices in the wfst framework," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4213–4216.