Training LSTM-RNN with Imperfect Transcription - Limitations and Outcomes

Martin Jenckel  
University of Kaiserslautern  
German Research Center for  
Artificial Intelligence (DFKI)  
Kaiserslautern, Germany  
martin.jenckel@dfki.de

Syed Saqib Bukhari  
University of Kaiserslautern  
German Research Center for  
Artificial Intelligence (DFKI)  
Kaiserslautern, Germany  
saqib.bukhari@dfki.de

Andreas Dengel  
University of Kaiserslautern  
German Research Center for  
Artificial Intelligence (DFKI)  
Kaiserslautern, Germany  
andreas.dengel@dfki.de

ABSTRACT
Bidirectional LSTM-RNN have become one of the standard methods for sequence learning, especially in the context of OCR due to its ability to process unsegmented data and its inherent statistical language modeling [5]. It has recently been shown that training LSTM-RNNs even with imperfect transcriptions can lead to improved transcription results [7, 14]. The statistical nature of the LSTM’s inherent language modeling can compensate for some of the errors in the ground truth and learn the correct temporal relations. In this paper we systematically explore the limits of the LSTM’s language modeling ability by comparing the impact of imperfect transcriptions with various hand crafted error types and real erroneous data created through segmentation and clustering. We show that training LSTM-RNN with imperfect transcriptions can produce useful OCR models even if the ground truth error is up to 20%. Further we show that it can compensate for some handcrafted error types with error rates of up to 40% almost perfectly.

CCS CONCEPTS
• Applied computing → Optical character recognition;

KEYWORDS
Historical Document Image Processing, OCR, Imperfect Transcription, LSTM, RNN

ACM Reference Format:

1 INTRODUCTION
Long-Short-Term Memory Recurrent Neural Networks (LSTM-RNN) were introduced in 1997 by Schmidhuber et. al. [6] and solved the existing problem of vanishing gradients when training RNN with gradient descent and back-propagation through time. They replaced the normal neuron with a more complex memory cell that can not just store information and therefore model long distance temporal correlations but also store the errors during the learning phase. Since then LSTM networks have been explored for OCR, offline and online handwriting recognition as well as many other sequence learning tasks like speech recognition or time series analysis [3, 4, 9, 13].

In recent work of [7, 14] it has been shown, that LSTM-RNNs are usable outside of the normal supervised training framework. They show that perfect annotations are not required, but LSTM-RNNs learn quite well even if the transcriptions they are trained on are imperfect, as they often are when generated automatically rather than crafted manually. Due to the nature of their inherent statistical language modeling LSTM-RNNs are able to learn and adjust as long as the error in the transcription is small enough and can produce results better than what they have been trained with. For LSTM-RNN it has been shown that, if trained on imperfect transcription with an 12.6% character error rate (CER), it can produce OCR models that reduce the error down to 6.7% [7]. In this paper we explore the statistical learning properties of LSTMs and find out at what level of ground truth error LSTMs no longer provide useful OCR models. The analysis focuses on errors generated via clustering, however some of these errors are also more general and have application to other methods of generating ground truth. First various error-types commonly observed during clustering are categorized in order to evaluate on how well LSTMs can cope with them by generating synthetic errors. For comparison the impact of realistic clustering errors is evaluated as well. Some of the work [8, 10] has been done for training deep neural networks offline and online handwriting recognition as well as many other related to OCR which is our main focus in this paper.

In section II of this paper various error types are introduced and analyzed with respect to LSTM training and clustering. In section III follows a detailed description of the used methodology followed by description of the experimental setup in section IV. The results are presented and discussed in Section 5. Section VI concludes the results and gives a short outlook towards future applications.

2 ERROR TYPES
First it is important to differentiate between errors that LSTM training can improve and which it can not improve.

Since real OCR data is not homogeneously distributed over all character classes, clustering, and sometimes human annotators, overlook classes with very few occurrences in the data. These characters will then be merged together with similar looking characters...
into a single cluster. One such example for Latin characters would be the character "ṣ" which gets grouped with either "e" or "g". The LSTM too has a problem with learning from very few examples and often groups it with similar characters as well. In this particular scenario the LSTM would not be supplied with the corresponding transcription and therefore could not learn or improve on these group of characters.

A second very similar case are characters which even the human eye might have a hard time differentiating. In historic Latin text for example there is an old form of the character "s" written as "f" which is similar in shape to an "f". If during clustering these two get mixed into the same cluster, then an LSTM will not be able to improve on this error, because again it would not be provided with the corresponding transcription.

On the other hand there are some error-types where the inherent statistical language modeling of LSTMs can compensate. For example in the case of "f" and "f" if during clustering some "f" are considered "f" instead or vice versa. We differentiate between one-sided and two-sided confusions. One-sided confusions mean only one character is sometimes confused as the other while two-sided confusion means both characters are confused with each other. Both error types are typical for many types of clustering but especially for algorithms with overlapping clusters or probabilistic models like Gaussian mixture model clustering.

As a third error type we consider random n-to-n confusion. Complete random confusions are typical for strongly degraded characters or scanning artifacts, where a character might become unidentifiable and therefore could be anything. To some degree this also models clustering errors where multiple one- and two-sided confusions between various characters are appearing simultaneously. Besides these single error types we also consider realistic clustering errors generated by iterative k-Means clustering as described in [7]. The resulting error is therefore a mixture of all before mentioned errors as well as additional errors introduced by over- and under-segmentation in the preprocessing. This can lead to either having additional characters (over-segmentation) or missing characters (under-segmentation).

Annotation errors can belong to either of the mentioned classes depending on how the data was annotated.

3 METHODOLOGY

3.1 Ground Truth Generation

In order to evaluate the impact of imperfect transcriptions, two protocols were followed when generating the erroneous ground truth. In the first protocol we artificially introduce different kinds of confusions in handcrafted annotation. For the one-sided and two-sided confusions, \( r_{ij} \in \{0, 1\} \) is the confusion rate of the character class \( J \) with the character class \( I \). This means that the total error \( r_{total} \) introduced into the ground truth is directly dependent on the size of the confused classes \( |J| \). For the one-sided (left) and two-sided (right) confusions this means:

\[
  r_{total} = \frac{r_{ij}|I|}{\sum_n |I_n|} \quad r_{total} = \frac{r_{ij}|I| + r_{ji}|J|}{\sum_n |I_n|}
\]  

(1)

For random confusion we looked at two different types of random n-to-n confusion. The first we call uniform random confusion. In this case the confusion rates are constant over all classes.

\[
  r_{ij} = \text{const.} = r_i \frac{1}{|J|}
\]

(2)

However this confusion is rather unrealistic. If all the confusion rates are the same, then large character classes are more likely to shrink and small character classes are more likely to increase in size, such that after applying the confusion, the distribution of character classes becomes more uniform. For this reason we consider a second confusion which we call proportional random confusion. In this case the chance for a character to be confused \( r_j \) is the same for all classes again, however the chance for any class \( J \) to be the confusion target, meaning the class the confused character is confused as, is chosen to be proportional to its size:

\[
  r_{ij} = r_j \frac{1}{|J|}
\]

(3)

This means large classes are more likely to be confusion targets and will keep the distribution over character classes almost the same. To evaluate realistic errors generated through clustering we followed the method described in [7]. After text line segmentation and clustering all clusters have been annotated manually in order to generate an imperfect transcription. We calculated the confusion rates between character classes by comparing the clustering results with the handcrafted ground truth. In order to evaluate the impact of increasing and decreasing errors we scaled the confusion rates while keeping the relative confusions the same.

The LSTMs innate language modeling ability is statistical in nature, meaning that confusion rates of 0.5 are where one expects to see a high drop in the LSTMs ability to compensate. For our evaluation we introduce confusion rates of 0.05 to 0.6 in 0.05 steps and total errors of 0.05 to 0.5 in 0.05 steps for realistic clustering and random n-to-n confusions.

3.2 OCRopus

For training the LSTM implementation of OCRopus is used. OCRopus is a complete OCR software package and its implementation is known for producing state of the art results while being easy to use [2, 11]. As part of the LSTM framework it provides a text line normalization method based on Gaussian filtering and affine transformation described in [15].

4 DATA

The used data is from a Latin version of the 15th century novel Narrenschiff. A sample image is shown in Figure 2. The dataset contains a total of xx binarized pages in text line form consisting of xx lines with xx individual characters. We chose about 80% of the data as our training set and split the rest is roughly equally sized validation and test set. The exact split can be seen in table 1. It should be noted that for the clustering the complete data set has

<table>
<thead>
<tr>
<th>pages</th>
<th>lines</th>
<th>chars</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>79</td>
<td>2673</td>
</tr>
<tr>
<td>val</td>
<td>10</td>
<td>349</td>
</tr>
<tr>
<td>test</td>
<td>11</td>
<td>396</td>
</tr>
</tbody>
</table>

Table 1: Overview over the used data set
Figure 1: One-Sided Confusion: Character Error Rates (CER) for 4 different one-sided confusions introduced to the training data. The first character is confused as the second. The absolute error in the training data (red), expected CER (green) and best model CER (blue) after training are compared. For almost all values the CER is smaller than the expected CER which means the LSTM could learn form the erroneous training data. The CER of the last trained model (cyan) is shown for one confusion and underlies stronger variance.

Figure 2: A Sample of the 15th century novel "Narrenschiff" in Latin script. Images are taken from the German government funded project, Kallimachos [1].

been used, as it is an unsupervised method and serves the purpose of identifying the character classes.

4.1 Parameters

The LSTM implementation in OCRopus requires some free parameters to be set. OCRopus only supports a single hidden layer whose size is set to 100. The height of the text line corresponds directly to the size of the input layer and has to be set for normalization. For the used data 48 is a reasonable value. Learning rate and momentum are kept at the default values of $1e^{-4}$ and 0.9 respectively. Each model was trained on 2673 text lines corresponding to ≈ 38 epochs; leading finally to 100,000 training iterations. OCRopus chooses training lines randomly, however changing to fixed training order did not change the results significantly. The same is true for using a fixed weight initialization. For a better comparison with real world application we used OCRopus default settings for training order and weight initialization.

5 RESULTS

During training a model has been recorded after every 1000 iterations. Based on the validation data the best model within the last 10 models has been chosen and is then evaluated on the test set. All reported character error rates CER have been calculated using the Levenstein-Distance. When introducing more than 2 confusions, the Levenstein-Distance will give lower error rates compared to the introduced confusion rates, so we upscaled the confusion to meet the desired CER.
5.1 Synthetic Confusions

For the one-sided and two-sided confusion we decided to look at three different confusion pairs that can be found regularly within the top 10 confusions when looking at either clustering or LSTM training of our data. In all three cases the class sizes are different. The first confusion to consider is between c and e. In many fonts the left curve of c and e is identical and some cs often get mixed in with the es in clustering. The confused character class c is with 2173 instances less than half as big as the confusion target class e with 5622 instances. The second confusion is between r and t. While these characters are not necessarily very similar, in some fonts, like the one in our data, slight degradations can easily lead to confusions between the two, e.g. the top part of the t missing or r having a slight extension to the left. In this scenario both classes have almost the same size with 3879 instances and 4333 instances respectively. The last pair we decided to use for a confusion are f and f. Even for a human it might be hard to differentiate between the two, even more so if the document is old and has various degradations. This time the confused character class f with 1523 instances is about three times the size of the confusion target class f with 531 instances. Even with high confusion rates the resulting introduced CER stays rather small, therefore the case where c and e as well as r and t are confused simultaneously is considered as well.

5.1.1 One-sided Confusions. Figure 1 shows the change in the CER (blue) for the various confusion rates and compares them to the expected CER (green) and the introduced CER in the ground truth (red). In all cases introducing confusions up to 40% show no significant increase in CER and at 45% a slight increase can be identified. After that the CER shows an upward trend and starts exceeding the expected CER at 55 – 60% confusion. The unexpectedly good result at 60% confusion for the c-e confusion can be considered an outlier. The magnitude of the resulting CER depends on the absolute class sizes, producing larger CER if the confused character class is bigger. One should note that for small confusion rates the expected increase in CER is less than 0.25%. In these cases the normal variance between models overshadows CER.

Figure 3: Two-Sided Confusion: Character Error Rates (CER) for 4 different two-sided confusions introduced to the training data. The absolute error in the training data (red), expected CER (green) and best model CER (blue) after training are compared. For almost all values the CER is smaller than the expected CER which means the LSTM could learn form the erroneous training data.
5.1.2 Two-sided Confusions. As shown in Figure 3 the CER curves look a bit different for two-sided confusions. For low confusion rates < 30% again the resulting CER (blue) stays around the established base line CER of 2.5 ± 0.25. In all 4 cases the CER starts increasing at about 45% again and only for confusion of 60% do we see it exceeding the expected CER.

Compared to the CER curves of the one-sided confusions the CER increases much faster for confusions higher than 45%. This could be due to the bigger CER introduced in the ground truth. In both cases the LSTM training shows no significant loss in accuracy for confusions up to 40%. Also in both cases training on the confused character classes starts to fail completely only for confusions of 55 – 60%.

With respect to the relative class size there is no conclusive performance difference between the selected confusion pairs. The same seems to be true for one-sided and two-sided confusions. This can be seen when comparing the two-sided c-e confusion (Figure 3 top left) with the one-sided r-t and c-e confusion (Figure 1 bottom right). The CER introduced in the ground truth in both cases has almost the same size and in both cases the CER behaves very similar and no significant differences can be identified.

5.1.3 Random n-to-n Confusion. All confusions considered so far are limited to few characters in otherwise correct ground truth. Real world applications rarely produces such isolated errors but introduces a whole assemble of different confusions at once. We simulate an assemble of confusions by considering two types of random n-to-n confusion as described in section 3.1.

Looking at the results in Figure 4 (bottom) it can be seen that in the proportional distribution the CER only increases very slowly resulting in low CER even for high introduced error. At the same time, uniformly distributing the confusion targets leads to a curve starting as flat but rising much faster as it approaches the 50% mark. The biggest difference between these two experiments is, that in the proportional case, the overall distribution over character classes is similar before and after the confusion. One can think of this as shuffling the characters within the data set, but keeping the overall distribution the same, as can be seen when comparing the green and blue graph in Figure 5. In the second case of uniform random confusion however the distribution of the character classes also becomes more uniform (compare the red graph in Figure 5). This could be compared to the humans ability to read words when some characters are shuffled compared to reading words when some characters are replaced. Additionally in both cases, random confusions destroy most long distance temporal relations, however in the case of uniform random confusions most short distance temporal relations are destroyed as well.

5.2 Realistic Clustering Error

To evaluate more realistic errors generated through clustering we followed the anyOCR pipeline as introduced by [7]. After segmentation, clustering and manual annotation the imperfect ground truth was generated. By keeping the relative confusion rates constant and up- and downscaling the original CER of 12.6% additional ground truths with varying CER from 5 – 50% are generated.

Figure 4: Clustering: Character Error Rates (CER) for realistic clustering and two types of random n-to-n confusions used to create training data. The absolute error in the training data (red), expected CER (green) and best model CER (blue) after training are compared. For almost all values the CER is smaller than the expected CER which means the LSTM could learn form the erroneous training data.

Figure 4 (top) shows the results for various levels of ground truth CER. It is notable, that for every level of CER the final result is lower than the introduced error, which is in line with the results presented in [7]. Compared to the random confusion the CER grows
learning the character distribution. If the character distribution in
the training data changes too much, then the LSTM fails to recog-
nize test data from the original distribution. If the distribution is
almost the same however, even at high error rates in the training
data, recognition is still good.

Overall the results show that LSTM-RNN is a powerful tool when
working with erroneous transcriptions and provide guidelines for
its application.

During out study we noticed that besides the LSTM’s ability to
learn from imperfect ground truth, the model selection based on
validation data plays an important role in the observed effect of
error reduction. The statistic nature of LSTM-RNN training allows
for good and bad models during the full training process and select-
ing the best model plays a major part in the overall performance of
the system.

Future research should focus on the implementation of these find-
ings into OCR pipelines or other LSTM based system. For ap-
proaches using synthetic data such as [12], training on synthetic
data could be improved if the same character distribution is pro-
vided within the training data.

ACKNOWLEDGMENTS

This work was partially funded by the BMBF (German Federal Min-
istry of Education and Research), project Kallimachos (01U14113C).

REFERENCES

Nurzagonien/Main

https://github.com/mldsdev/ocropus

Forget. Continual Prediction with LSTM. 12, 10 ([n. d.]), 2451–2471.

Biologically plausible speech recognition with LSTM neural nets. In Interna-
tional Workshop on Biologically Inspired Approaches to Advanced Information

[5] Graves, Alex and Fernández, Santiago and Gomez, Faustino J. and Schmidhuber,
Jürgen. 2006. Connectionist temporal classification: Labelling Unsegmented
Sequence Data with Recurrent Neural Networks. In IJICML’06.


Sequence Learning Based OCR System for Unlabeled Historical Documents. In

//arxiv.org/abs/1705.10694

evaluation and (semi) automatic improvement of mixed models for OCR on
05157

[10] Sainbayar Sukhbaatar and Rob Fergus. 2014. Learning from Noisy Labels with

for Printed English and Fraktur using LSTM Networks. In ICDAR. Washington
D.C. USA.

for Medieval Scripts. In 2nd International Conference on Natural Sciences and

OCR for Printed English and Fraktur using LSTM Networks. (08 2013).

A Sequence Learning OCR System Trained on Isolated Characters. In The 12th

1D and 2D LSTM Architectures for Recognition of Handwritten Arabic. In
DMR-XXI. USA.

6 DISCUSSION AND OUTLOOK

The results allow for the conclusion of three major points. First
from the comparison of one-sided and two-sided confusions we
conclude that LSTM can handle both confusions types up to 40%
almost perfectly. In the case of clustering this also means, that con-

fusion are not the main problem, as long as they are small enough.

The second conclusion comes from the comparison of the two types
of random n-to-n confusions. Especially at high confusion rates the
difference in performance suggests that LSTM training relies on

Figure 5: Character distributions: Distributions after gener-
ating the ground truth for the uniform random n-to-n confu-
sion at 50% CER (green), the proportional random n-to-n con-
fusion at 50% CER (red) and the upscaled clustering results at
50% CER (black) compared to the correct ground truth (blue).
The last character class represents missing characters.