A Two-Step Approach for Automatic OCR Post-Correction

Staatsbibliothek zu Berlin
Preußischer Kulturbesitz

Robin Schaefer & Clemens Neudecker
Staatsbibliothek zu Berlin - Preußischer Kulturbesitz
{firstname.lastname}@sbb.spk-berlin.de

1. Introduction

Motivation: The quality of Optical Character Recognition (OCR) is a key factor in the digitisation of historical documents. OCR errors are a major obstacle for downstream tasks and have hindered advances in the usage of the digitised documents.

A post-correction pipeline has the following tasks:
1. To correct OCR errors.
2. To ignore the already correct data.

Previous work: OCR post-correction has been approached using techniques like statistical language modelling [1], Anagram Hashing [2] and Statistical/Neural Machine Translation [3,4]. Experiments showed, that post-correction becomes more difficult as the Character Error Rate (CER) decreases.

Our approach: We propose an alternative two-step pipeline for OCR post-correction consisting of the following components:
1. **Detector**: Reads OCRed sequence and decides if error exists. Forwards sequence to *translator* only if it is erroneous.
2. **Translator**: Reads sequence declared as erroneous by *detector* and corrects errors.

Benefits of the two-step approach:
(i) By decreasing the proportion of correct OCRed data, the CER fed into the translator is increased.
(ii) By excluding correct sequences from translation, we can avoid to insert additional errors.

2. Data [i]

Using OCR-D [5, ii] we created two data sets based on historical German works (Deutsches Textarchiv, “German Text Archive”) (17th - 19th century).

Processing steps:
1. Aligned lines of GT and OCRed documents using dinglehopper [iii].
2. Calculated CER for each OCR sequence.
3. Removed line pairs if CER > 10%.
4. Applied sliding window approach (4 tokens per window).
5. Removed non-German sequence pairs.

Data sets:
(T = translator; D = detector)

	Set I	Set II
Used for training of ...	T (One-Step) & D (Two-Step)	T (Two-Step)
Training	365,000	196,000
Validation	56,800	22,000
Testing	56,800	22,000

SPONSORED BY THE Federal Ministry of Education and Research
Qurator Curation Technologies

3. The Standard Approach (One-Step)

First, we approached OCR post-correction using a standard sequence-to-sequence model.

The model has two LSTM-based components:
- **Encoder**: Reads the OCRed sequence and derives a matrix representation.
- **Decoder**: Reads the matrix representation and converts it to the correct(ed) output sequence.

Hyperparameters and other design choices:
- Decoder: based on attention component
- Hidden node size: 256
- Number of layers: 1
- Epochs: 970
- Learning rate: 0.0001
- Batch size: 200

Results:

Approach	CER (pre)	CER (post)
One-Step	1.2%	1.6%

The low CER of the dataset seems to be difficult to correct for the one-step approach.

4. The Two-Step Pipeline [iv]

Detector: An LSTM model that outputs for every encoding probabilities of it being correct or incorrect.

Training objective: Achieve a high precision and a low false positive rate, i.e. a small number of sequences wrongly classified as erroneous.

Hyperparameters:
- Hidden node size: 512
- Number of layers: 3
- Epochs: 138
- Learning rate: 0.0001
- Batch size: 200

Results:

F1	Precision	Recall
81%	90%	74%

Confusion matrix: (pos = incorrect; neg = correct)

	Predicted neg.	Predicted pos.
Target neg.	41386	1123
Target pos.	3730	10561

In accordance with our training objective, we received a high precision (see confusion matrix: Predicted pos.). About 90% of sequences are correctly classified as incorrect.

Translator: Hyperparameters and architecture are identical to the one-step model. Model was trained on data set II, which contains 90% incorrect sequences (in accordance with the detector results).

Results:

Approach	CER (pre)	CER (post)
Two-Step	4.3%	3.6%

5. Applying the Full Pipeline

We applied the full pipeline on both test sets. No correct sequences were removed manually.

Results (Detector):

F1	Precision	Recall
79%	87%	72%

19,800 sequences were classified as erroneous and forwarded to the translator.

Results (Translator):
(NEI = new errors introduced)

Approach	CER (pre)	CER (post)	NEI
Two-Step	1.1%	0.9%	0.3%
One-Step	1.1%	2.1%	6%

6. Conclusion

The results confirm the benefits of our approach.

Inserting the detector...
1. ...substantially increases the translation results. (1.1% to 0.9% vs 2.1%)
2. ...substantially decreases the number of newly introduced errors. (NEI: 0.3% vs 6%)

Outlook:
- Improve the decoder’s attention mechanism.
- Experiment with alternative translation architectures (e.g. Generative Adversarial Nets)

Resources & References

Data & Code:
[i] Data (two-step approach): `https://zenodo.org/communities/stabi/`.
[ii] OCR-D implementation: `https://github.com/qurator-spk/ocrd-galley`.
[iii] dinglehopper: `https://github.com/qurator-spk/dinglehopper`.
[iv] Code (two-step approach): `https://github.com/qurator-spk/sbb_ocr_postcorrection`.

References:
[1] Tong & Evans (1996). *A Statistical Approach to Automatic OCR Error Correction in Context*.
[2] Reynaert (2008). *Non-interactive OCR post-correction for giga-scale digitization projects*.
[3] Amrhein & Clematide (2018). *Supervised OCR error detection and correction using statistical and neural machine translation methods*.
[4] Rigaud, Doucet, Coustaty & Moreux (2019). *ICDAR 2019 Competition on post-OCR text correction*.
[5] Neudecker, Baierer, Federbusch, Boenig, Würzner, Hartmann & Herrmann (2019). *OCR-D: An end-to-end open source OCR framework for historical printed documents*.