Neural OCR Post-Hoc Correction of Historical Corpora

Lijun Lyu[1], **Maria Koutraki**[1], **Martin Krickl**[2], **Besnik Fetahu**[1,3]

[1]L3S Research Center, Leibniz University of Hannover / Hannover, Germany
[2]Austrian National Library / Vienna, Austria
[3]Amazon / Seattle, WA, USA
lyu@L3S.de, koutraki@L3S.de, martin.krickl@onb.ac.at, besnikf@amazon.com

Abstract

Optical character recognition (OCR) is crucial for a deeper access to historical collections. OCR needs to account for *orthographic* variations, *typefaces*, or *language evolution* (i.e., new *letters*, *word spellings*), as the main source of *character*, *word*, or *word segmentation* transcription errors. For digital corpora of historical prints, the errors are further exacerbated due to low scan quality and lack of language standardization.

For the task of OCR post-hoc correction, we propose a neural approach based on a combination of recurrent (RNN) and deep convolutional network (ConvNet) to correct OCR transcription errors. At character level we flexibly capture errors, and decode the corrected output based on a novel attention mechanism. Accounting for the input and output similarity, we propose a new loss function that rewards the model's correcting behavior.

Evaluation on a historical book corpus in German language shows that our models are robust in capturing diverse OCR transcription errors and reduce the word error rate of 32.3% by more than 89%.

1 Introduction

OCR is at the forefront of digitization projects for cultural heritage preservation. The main task is to *identify* characters from their *visual* form into their *textual* representation.

Scan quality, *book layout*, *visual* character similarity are some of the factors that impact the output quality of OCR systems. This problem is severe for historical corpora, which is the case in this work. We deal with *historical* books in German language from the 16th–18th century, where characters are added or removed (e.g., *long s* – ſ), word spellings change (e.g., *''vnd''* vs. *''und''*) that often lead to word and character transcription errors. Figure 1 shows examples pages conveying the complexity of this task.

There are several strategies to correct OCR transcription errors. *Post-hoc correction* is the most common setup (Dong and Smith, 2018; Xu and Smith, 2017). The input is an OCR transcribed text, and the output is its corrected version according to the error-free ground-truth transcription. For instance, Dong and Smith (2018) use a *multi-input attention* to leverage *redundancy* among textual snippets for correction. Alternatively, domain specific OCR engines can be trained (Reul et al., 2018a), by using manually aligned *line image segments* and *line text* (Reul et al., 2018b). However, manually acquiring such ground-truth is highly expensive, and furthermore, typically, historical corpora do not contain redundant information. Moreover, each book has its own characteristics—*typeface* styles, *regional* and *publisher's* use of language, and so forth.

In this work, we propose a post-hoc approach to correct OCR transcription errors, and apply it to a historical collection of books in German language. As input we have only the OCR transcription of book from their scans, for which we output the *corrected* text, that we assess with respect to the ground-truth transcription carried out by human annotators without any spelling change, language normalization, or any other form of interpretation. By considering only the textual modality for our approach, we provide greater flexibility of applying our approach to historical collections where the image scans are not available. However, note that since orthography was not standardized, there can be *parallel* spellings of the *''same''* word (e.g., *''und''* vs. *''vnd''*) within the same book, which may pose challenges for approaches that use the text modality only.

Our approach, **CR**, consists of an encoder-decoder architecture. It encodes the erroneous

Transactions of the Association for Computational Linguistics, vol. 9, pp. 479–493, 2021. https://doi.org/10.1162/tacl_a_00379
Action Editor: Sebastian Padó. Submission batch: 9/2020; Revision batch: 1/2021; Published 4/2021.

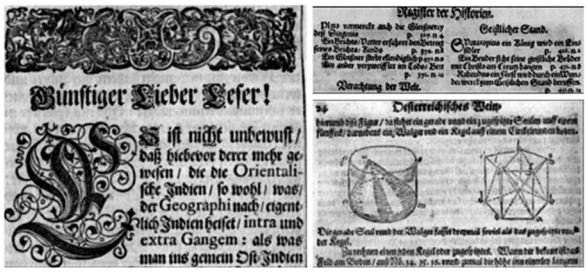

Figure 1: Pages with coexisting typefaces (*Fraktur* and *Antiqua*), double columns, and images surrounded by texts.

input text at *character level*, and outputs the corrected text during the decoding phase. Representation at character level is necessary given that OCR transcription errors at the most basic level are at character level. The input is encoded through a combination of RNN and deep ConvNet (LeCun et al., 1995) networks. Our encoder architecture allows us to flexibly encode the erroneous input for post-hoc correction. RNNs capture the *global input context*, whereas ConvNets construct local *sub-word* and *word compound* structures. During decoding the errors are corrected through an RNN decoder, which at each step through an attention mechanism combines the RNN and ConvNet representations and outputs the corrected text.

Finally, since the input and output snippets are highly similar, loss functions like *cross-entropy* lean heavily towards rewarding *copying* behavior. We propose a custom loss function that rewards the model's ability to *correct* transcription errors.

In this work, we make the following contributions:

- a data collection approach with a parallel corpus of 800k sentences from 12 books (16th–18th century) in German language;
- an error analysis, emphasizing the diversity and difficulty of OCR errors;
- an approach that flexibly captures erroneous transcribed OCR textual snippets and robustly corrects character and word errors for historical corpora.

2 Related Work

Redundancy Based. The works in Lund et al. (2013), Lund et al. (2011), Xu and Smith (2017), and Lund et al. (2014) view the problem of post-hoc correction under the assumption of *redundant* text snippets. That is, multiple redundant text snippets are combined and under the majority voting scheme the correction is carried out. Dong and Smith (2018) propose a multi-input attention model, which uses redundant textual snippets to determine the correct transcription during the training phase. While there is redundancy for contemporary texts, this cannot be assumed in our case, where only the OCR transcriptions are available. Our approach can be seen as complementary to data augmentation techniques that exploit redundancy.

Rule Based Correction. Rule based approaches compute the edit cost between two text snippets based on weighted finite state machines (WFSM) (Brill and Moore, 2000; Dreyer et al., 2008; Wang et al., 2014; Silfverberg et al., 2016; Farra et al., 2014). WFSM require predefined rules (*insertion, deletion,* etc., of characters) and a lexicon, which is used to assess the transformations. The rewrite rules require the mapping to be done at the word and character level (Wang et al., 2014; Silfverberg et al., 2016). This process is expensive and prohibits learning rules at scale. Furthermore, lexicons are severely affected by out-of-vocabulary (OOV) problems, especially for historical corpora. A similar strategy is followed by Barbaresi (2016), who uses a spell checker to detect OCR errors and generate correction candidates by computing the edit distance. OCR transcription errors are highly contextual and there are no one-to-one mappings of misrecognized characters that can be addressed by rules (cf. Figure 6).

Machine Translation. Post-hoc correction can also be viewed as a special form of machine translation (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014). For post-hoc correction of OCR transcription errors, the only reasonable representation is based on characters. This is due to the character errors and word segmentation issues, which can only be detected when encoding the input text at character level. Results from *spelling correction* (Xie et al., 2016) and *machine translation* (Ling et al., 2015; Ballesteros et al., 2015; Chung et al., 2016; Kim et al., 2016; Sahin and Steedman, 2018) indicate that character based models perform the best. Methods based on statistical machine translation (SMT) (Afli et al., 2016) use a combined set of features at word level and language models for post-hoc correction. Schulz and Kuhn (2017) use a multi-modular

ID	Barcode	Year	Author	Location	Layout	pages	WER	CER
1	Z165780108	1557	H. Staden	Marburg	single column	177	66.8%	16.3%
2	Z205600207	1562	M. Walther	Wittenberg	single column	75	54.1%	12%
3	Z176886605	1603	B. Valentinus	Leipzig	single column	134	46.8%	14.3%
4	Z185343407	1607	W. Dilich	Kassel	single column	313	60.8%	17.1%
5	Z95575503	1616	J. Kepler	Linz	single column w. pg. margin	119	59%	17.2%
6	Z158515308	1647	A. Olearius	Schleswig	single column	600	51.8%	17.7%
7	Z176799204	1652	S. von Birken	Nürnberg	single column	190	55.9%	13.8%
8	Z165708902	1672	J. Jacob Saar	Nürnberg	single column w. pg. margin	186	33%	10.4%
9	Z22179990X	1691	S. von Pufendorf	Leipzig	single column	665	32.7%	7.7%
10	Z172274605	1693	A. von Schönberg	Leipzig	single column	341	67.6%	30.5%
11	Z221142405	1699	A. a Santa Clara	Köln	single/double column w. pg. margin	794	51.4%	16.1%
12	Z124117102	1708	W. Bosman	Hamburg	single column	601	37.8%	6.7%

Table 1: Detailed book information can be accessed from the *ÖNB* portal using the barcode.

approach combining dictionary lookup and SMT for word segmentation and error correction. However, the dataset used for training is limited to books of the same topic, and requires manual supervision in terms of feature engineering.

Sequence Learning. As is shown later, character based RNN models (Xie et al., 2016; Schnober et al., 2016) are insufficient to capture the complexity of compound-rich languages like German. Alternatively, ConvNets have been successfully applied in sequence learning (Gehring et al., 2017b,a). Although the performance of ConvN et alone is insufficient for post-hoc correction, we show that their combination yields optimal post-hoc correction performance.

OCR Engines. Slightly related are the works of Reul et al. (2018a,b), which retrain OCR engines on a specific domain. The assumption is that clean *line scans* with the same fontface are available. In this way, the trained OCR engines are more robust in transcribing text scans of the same fontface. Figure 1 shows that this is rarely the case, and many characters induce orthographic ambiguity. Furthermore, in many cases the OCR process is unknown, with image scans being the only material available.

3 Data Collection & Ground-Truth

In this section, we describe our data collection efforts and the ground-truth construction process. Currently, there is no large-scale historical corpus in German language that can be used for post-hoc correction of OCR transcribed texts. The collected corpus and constructed ground-truth of more than 854k pairs of OCR transcribed textual snippets and their corresponding manual transcriptions, together with the source code are available.[1]

3.1 Book Corpus

We first describe the process behind selecting our corpus of historical books in German language. As our input textual snippets for OCR post-hoc correction we consider the publicly available historical collection of transcribed books, which are freely accessible by the *Austrian National Library* (OeNB).[2] The transcription of books from their image scans is done in partnership with Google Books project, which uses Google's proprietary OCR frameworks. Given that this process is an automated process, the transcriptions are not error free.

For the ground-truth transcriptions we turn to another publicly available collection, namely, *Deutsches Textarchiv* (DTA).[3] It contains manually transcribed books based on community efforts. The transcriptions are error free and as such are suitable to be used as our ground-truth. We consider the overlap of books present in both DTA and OeNB, providing us with the erroneous input textual snippet from OeNB and the corresponding target error-free transcription from DTA.

Table 1 shows our books corpus, consisting of the *overlap* between these two repositories,

[1] https://github.com/GarfieldLyu/OCR_POST_DE.
[2] https://www.onb.ac.at/.
[3] http://www.deutschestextarchiv.de/.

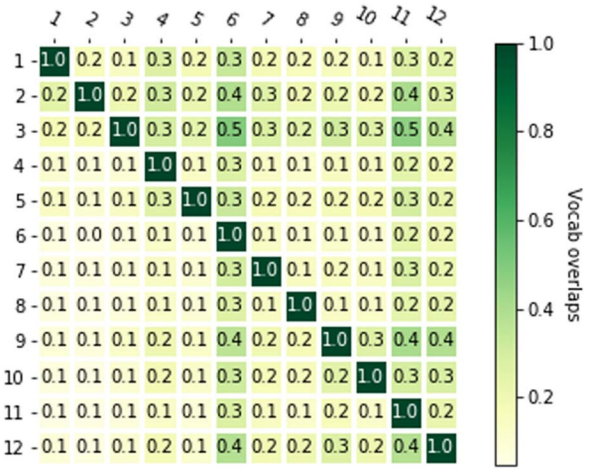

Figure 2: Vocabulary overlap between books.

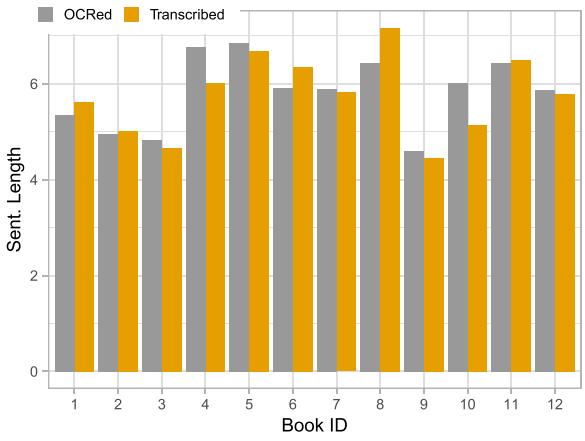

Figure 3: Sentence length distributions.

with 12 books in *German language* from the *16th–18th* centuries. Understandably, considering the publication period, there is little overlap across the different books. Figure 2 shows the *vocabulary* overlap between books, which on average is around 20%. This presents an indicator of a corpus with high diversity and low redundancy, representing a realistic and challenging evaluation scenario for post-hoc correction.

3.2 Ground-Truth Construction

The constructed ground-truth consists of the mapped OCR transcribed text to their manually transcribed counterparts, resulting in a parallel corpus of OCRed *input* text and the *target* manually transcribed counterparts.

To construct the parallel corpus is challenging. OCR transcribed books contain all pages (e.g., *content* and *blank* pages), while the manually transcribed books keep only the *content* pages. Furthermore, books are typically transcribed line by line by OCR systems, which often fail to detect page layout boundaries (multi-column layouts or printed margins). Therefore, accurate ground-truth construction even at page level is challenging.

An important aspect is the *granularity* of parallel snippets. Figure 3 shows the average *sentence length* distribution for OCR and manually transcribed books. We consider sentences, which are demarcated by the symbol *''/''*, when this information is not available we fall back to text lines. The average sentence length is 5–6 tokens, with an average of up to 100 characters.

Therefore, we consider snippets of *5 tokens* for mapping, as longer ones (e.g., paragraphs), are highly error prone. Furthermore, depending on scan quality, page content (e.g., if it contains figures or tables), the error rates from OCR transcriptions can vary greatly from page to page, making it impossible to consider lengthier snippets for the automated and large-scale ground-truth construction.

To construct an accurate ground-truth for OCR post-hoc correction, we propose the following two steps: (i) *approximate matching*, and (ii) *accurate refinement*.

3.2.1 Approximate Snippet Matching

From the OCR transcribed books, we generate textual snippets of 5 tokens length and compute approximate matches to snippets of 5–10[4] tokens from the manually transcribed books. Approximate matching at this stage is required for two reasons: (i) text lines from OCR and manually transcribed books are not *aligned* at line level in the books, and (ii) an exhaustive pair-wise comparison of all possible snippets of length 5 is very expensive.

We rely on an efficient technique known as *locality sensitive hashing* (LSH) (Rajaraman and Ullman, 2011) to put textual snippets that are *loosely* similar into the same bucket, and then based on the Jaccard similarity determine the highest matching pair. The hashing signatures and the Jaccard similarity are computed on *character tri-grams*.

The resulting mappings are not error free, and often contain *extra* or *missing* words. Such errors are introduced often due to the OCR engines

[4]Lengthier snippets are necessary due to segmentation errors, resulting in longer snippets.

```
fen angenommen werden als in-def----------
    |||||||||||||||||||||||| ||
----angenommen werden als in defWillkühr
```

Figure 4: Ideal textual snippet alignment. Gap characters ‘‘-’’ and additional characters are removed from both sides.

breaking over the multi-column layouts of books, inclusion of table/figure captions, word segmentation errors (under or over segmentation). Snippets from OCR transcriptions that do not have a matching above a threshold (< 0.8) are dropped.

Matching Coverage: Finally, to ensure that our ground-truth construction approach does not severely affect coverage of the matched pairs, we conduct a manual analysis of two books with different layouts (books ID 6 and 11, cf. Table 1) for 10 randomly selected pages from each book. For book 6, which has good scan quality, for snippets of 5 tokens, we are able to find a relevant match from the manual transcription on average for 270 out of 300 snippets per page. The dropped snippets in absolute majority of the cases consist of footnotes or page headings. In the case of book 11, which has a bad scan quality and is of double column layout, from 400 snippets, only 200 have a match. Upon inspection, we find that this is mostly due to the erroneous transcription by OCR systems, which mistakenly merges lines from different columns into a single line. These snippets are corrupted, and cannot be matched to snippets extracted from the manually transcribed books.

3.2.2 Accurate Refinement

The main issue with the approximate matching through LSH, is that there are *extra* words appearing at the head or end of either the input or output snippets. The extra tokens stem mostly from snippets that match lengthier or shorter ones due to word segmentation errors. Such *additional/missing* words are not desirable, and thus, in this stage we refine the above snippet mappings. We perform a *local pairwise sequence alignment*[5] that finds the best matching local sub-snippets. The remaining extra characters are removed (e.g., tokens *‘‘fen’’* and *‘‘Willkühr’’* are removed as they are not part of a local alignment).

[5] https://biopython.org/DIST/docs/api/Bio.pairwise2-module.html.

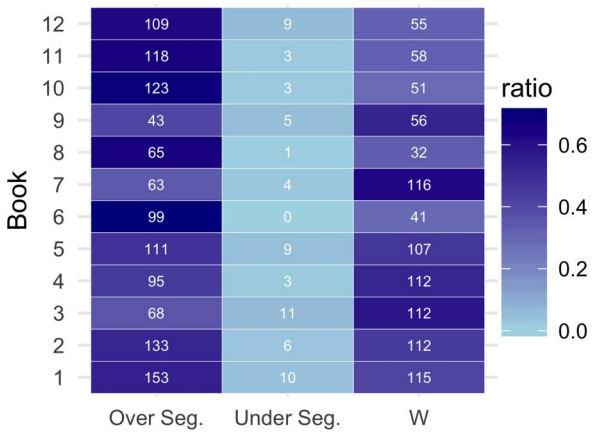

Figure 5: Book OCR error type distribution.

4 Data Analysis

Based on a *manual* analysis of a random sample of 100 snippet pairs taken from each book from our ground-truth, we analyze the various OCR transcription error types.

This is a crucial step towards developing post-hoc correction models in a systematic manner. OCR errors are highly contextual and are dependent on several factors, and as such there are no one-to-one rules that can be used to correct OCR errors. Furthermore, these errors are increased when dealing with historical corpora, as fontfaces, book layouts, and language use are highly unstandardized.

We differentiate between the following errors: (i) **over-segmentation** is an error when multiple words are merged into one, (ii) **under-segmentation** when a word is split into two, and (iii) **word error**, typically caused by *misrecognized characters*, converting it into an *invalid word* or changing its meaning to a *different valid word*.

4.1 Error Types and Distribution

Figure 5 shows an overview of the *error types* for the different books in our corpus.

Over-segmentation is one of the most common OCR transcription errors, with 54% of the cases. The errors often arise due to OCR systems misrecognizing spaces between words and characters in a word, as these are often not clearly distinguishable. These errors are challenging since the words may represent valid words, which is an even more challenging problem for *compound* rich languages like German.

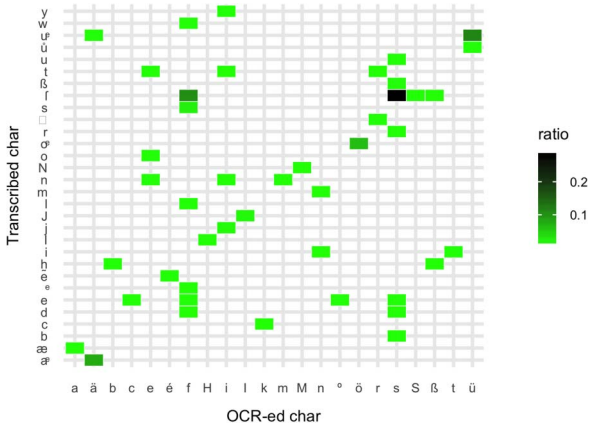

Figure 6: The misrecognized character (x–axis) and their valid transcriptions (y–axis).

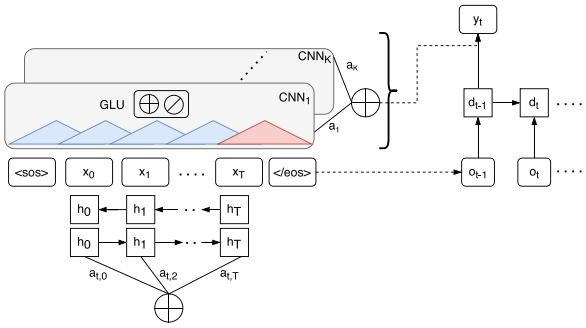

Figure 7: Approach overview.

Under-segmentation errors are less common (with 3%), and are mostly due to line-breaks and book layouts.

Word-errors represent the second most frequent OCR error category with 43%. These errors are often caused due to the *orthographic* visual similarity between characters, thus resulting in invalid words or changing the word's meaning altogether. Other relevant factors are the scan quality or book layouts.

Figure 6 shows that word errors are contextual, with no simple mappings between misrecgonized characters. An indicator that they are not solely due to visual character similarity, as they are often misrecognized to completely different characters.

5 Neural OCR Post-Hoc Correction

Figure 7 shows an overview of our encoder-decoder architecture for post-hoc OCR correction. At its core, the encoder combines RNN and deep ConvNets for representation of the *erroneous* OCR transcribed snippets at the *character* level. During the decoder phase an RNN model corrects the errors one character at a time by using an attention mechanism that combines the encoder representations, a process repeated until an end of a sentence is encountered.

5.1 Encoder Network

We encode the erroneous OCR snippets at the character level for three reasons. First, word representation is not feasible due to *word errors*. Second, only in this way can we capture *erroneous characters*. Finally, we avoid OOV issues, as there are no vocabularies for historical corpora.

The encoder consists of a RNN and a deep ConvNet network. The intuition is that RNNs capture the *global context* on how OCR errors are situated in the text, while deep ConvNets capture and enforce *local sub-word/phrase* context. This is necessary for *word segmentation* errors, which might bias RNN models towards more frequent tokens (e.g., *''alle in''* vs. *''allein''*).[6]

5.1.1 Recurrent Encoder

First, we apply a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) that reads the erroneous OCR snippet $X = (x_1, \ldots, x_T)$, encoding it into $h_T = \left[\overrightarrow{LSTM}(X), \overleftarrow{LSTM}(X)\right]$.

We use recurrent models because of their ability to detect *erroneous* characters and to capture the *global context* of the OCR errors. In § 4 we showed that for most of the *erroneous* characters, the target *transcribed characters* vary, a variation that can be resolved by the general context of the snippet.

Finally, we will use h_T to conditionally initialize the decoder, since the *input* and *output* snippets are highly similar.

5.1.2 Convolutional Encoder

We find that 57% of errors are *word segmentation* errors. Often, these errors have a local behavior, such as merging or splitting words. While in theory this information can be captured by the RNN encoder, we notice that they are biased towards frequent sub-words in the corpora with tokens being wrongly split or merged.

We apply deep ConvNets to capture the *local context* (i.e., *compound* information) of tokens. ConvNets through their kernels limit the influence that characters beyond a token's context may have

[6]Both tokens are correct, with the first being more frequent.

in determining whether the subsequent decoded characters forming a token should be split or merged.

We set the kernel size to 3 and test several configurations in terms of ConvNet layers, which we empirically assess in § 7.2. Since we are encoding the OCR input at character level, determining the right granularity of representation is not trivial. Hence, the multiple layers l will flexibly learn from *fine* to *coarse* grained representation of the input. The learned representation at layer l is denoted as $h^l = \left(h_1{}^l, \ldots, h_T{}^l\right)$. In between each of the layers, we apply *non-linearity* such as gated linear units (Dauphin et al., 2017) to control how much information should pass from the bottom to the top layers.

5.2 Decoder Network

The decoder is a single LSTM layer, which generates the corrected textual snippet one character at a time. We *initialize* it with the last hidden state from the BiLSTM encoder h_T, that is, $o_1 = h_T$ in Equation (1), which biases the decoder to generate sequences that are similar to the input text.

$$p\big(o_i|o_{i-1}, \ldots, o_1, \mathbf{x}\big) = g\big(o_{i-1}, d_i, c_i\big) \quad (1)$$

where d_i is the current hidden state of the decoder, and o_{i-1} represents the previously generated character. c_i is the context vector from the encoded OCR input snippe, which combines the RNN and deep ConvNet input representations through a multi-layer attention mechanism, which we explain below.

5.2.1 Multi-layer Attention

Using jointly RNNs with deep ConvNets as encoders allows for greater flexibility in capturing the complexities of OCR errors. Furthermore, the multi-layers of the ConvNets capture from fine to coarse grained local structures of the input. To harness this encoding flexibility, we compute the context vector c_i for each decoder step d_i as following.

First, for each decoder state d_i at step i, we compute the weight of the representations computed by the deep ConvNet at the different layers. The weights, computed in Equation 2, correspond to the *softmax* scores, which are computed based on the dot product between d_i and the hidden layers h_j^l from the l layers of the ConvNet.

$$a_{ij}^l = \frac{\exp(e_{ij}^l)}{\sum_{k=1}^{T} \exp(e_{ik}^l)};\ e_{ij}^l = d_i \cdot h_j^l \quad (2)$$

At each layer l in the ConvNet encoder, the attention weights assess the importance of the representations at the different granularity levels in correcting the OCR errors during the decoder phase. To compute c_i, we combine the RNN and deep ConvNet representations, as scaled by the attention weights as following:

$$c_i = \frac{1}{L} \sum_{l=1}^{L} \sum_{j=1}^{T} a_{ij}^l \cdot [h_j^l, h_j] \quad (3)$$

5.3 Weighted Loss Training

Conventionally, encoder-decoder architectures are trained using the cross-entropy loss, $\mathcal{L} = -P_{tgt} \cdot \log P_{pred}$, with P_{tgt} and P_{pred} being the *target* and the *predicted* probability distributions of some discrete vocabulary.

For OCR post-hoc correction, cross-entropy does not properly capture the nature of this task. Models are biased to simply *copy* from input to output, which in this task represent the majority of cases. In this way, failure at correcting erroneous characters diminish, as all time-steps are treated equally. We propose a weighted loss function that rewards higher models for their *correcting* behavior. The modified loss function is shown in Equation 4.

$$\mathcal{L}' = \mathcal{L} \cdot \big(1 - \lambda P_{src} \cdot P_{tgt}\big);\ \ 0 < \lambda < 1 \quad (4)$$

The new loss function combines the cross-entropy loss $\mathcal{L}$ and an additional factor that considers the source and target characters. The second part of the equation captures the amount of desirable *copying* from input to output. If the input and output characters are the same, then $P_{src} \cdot P_{tgt}$ yields 1, otherwise 0, where P_{src} and P_{tgt} are one-hot character encodings of the input and output snippets. λ controls by how much we want to dampen this behavior. $\mathcal{L}'$ rewards higher the model's ability to correct erroneous sequences.

6 Experimental Setup

In this section, we introduce the experimental setup and the competing methods for the task of post-hoc OCR correction.

6.1 Evaluation Scenarios

According to our error analysis in § 4 and the highly diverging vocabularies across books (cf. § 3.1), we distinguish two evaluation scenarios. Here we use part of the ground-truth, where we select instances by first sampling pages from the books, namely the instance pairs coming from the sampled pages.

We assess the performance of models for two significant factors that may impact their correction behavior: (i) *eval–1* assesses the model's post-hoc correction behavior on unseen OCR transcription errors related to the book source and publication date, and diverging book content (cf. Figure 2), and (ii) *eval–2* tests the impact on correction performance when models have encountered all OCR errors based on random sampling.

eval–1: We split the data along the temporal axis, with training instances coming from books from the 16th and 18th centuries, and test instances from the 17th century. This scenario is challenging as there are diverging error types due to scan quality, and other orthographic variations related to the publishers and other book characteristics. The 17th century books have more diverse errors, as there are more books, and the initial OCR transcription error rates are higher.

We use 70% of the data for training, and 10% and 20% for validation and testing, with 269k, 27k, and 89k instances respectively.

eval–2: We randomly construct the training, validation, and testing splits, thus ensuring that the models have observed all error types, which should result in better post-hoc correction behavior. Furthermore, contrary to *eval–1*, where the splits are dictated by the publication date of the books, in this case, we use slightly different splits for training, validation, and testing. We use 65%, 10%, and 25%, for training, validation, and testing, respectively. The absolute number is 417k, 42k, and 166k, respectively.

6.2 Evaluation Metrics

To assess the post-hoc correction performance of the models, we use standard evaluation metrics for this task: (i) word error rate (**WER**), and (ii) character error rate (**CER**). The error rates measure the number of word/character *substitutions*, *insertions*, and *deletions*, normalized by the *total length* of the transcribed sequence, in characters for **CER** and number of words for **WER**.

6.3 Baselines

In the following we describe the approaches we compare against. In all cases, the input is represented at character level with 128 embedding dimensions. The cell units (i.e., LSTMs and ConvNets) are of 256 dimensions.

CH: Xie et al. (2016) use an RNN model for *spelling correction*, a task slightly similar to OCR post-hoc correction. Yet, the error types and their distribution are of a different nature. CH is a standard attention based encoder-decoder (Bahdanau et al., 2015), that corresponds to our CR model without ConvNets and the custom loss function.

CH$_\lambda$: To assess the impact of the introduced loss function, we train CH with the custom loss (cf. § 5.3). The optimal λ is set based on the validation set. This presents the ablated model of our approach CR without the ConvNet encoder and mult-layer attention.

PB: Cohn et al. (2016) propose a symmetric attention mechanism for RNN based encoder-decoder models. That is, encoder and decoder timesteps are strongly aligned. A similar alignment between input and output is expected for this task.

Transformer: By pretraining on large corpora, Transformers have (Vaswani et al., 2017) achieved the state-of-the-art results in various NLP tasks. In our case, pretraining on historical corpora is not possible due to the scarcity of such data, while pretraining on contemporary German corpora did not show any improvement. The self-attention mechanism is highly flexible in capturing intra-input and input-output dependencies, which is very important for post-hoc correction. We use the implementation in TK (Gagnon-Marchand and LJQ., n.d.) with 3 layers and 8 attention heads, and 512 dimensions for the output model, and encode input at character level.

Other Approaches: ConvSeq (Gehring et al., 2017b), part of our encoder network, yields performance below all the other competitors, hence we do not include its results here. Similarly, rule-based models based on FST (Silfverberg et al., 2016) yield poor performance. We believe this is

due to the inability to establish one-to-one mapping of rules for correction, and the requirement for valid word vocabularies.

6.4 CR: Approach Configuration

For our approach **CR**, based on a validation set, the number of ConvNet layers is set to $k = 1$ and $k = 3$, and set $\lambda = 0.3$ and $\lambda = 0.1$, for *eval–1* and *eval–2*, respectively.

7 Evaluation

In this section, we provide a detailed evaluation discussion and discuss limitations.

1. Post-hoc correction evaluation as measured through WER and CER metrics.
2. Ablation study for our approach CR.
3. Performance of CR for post-hoc correction at page level.
4. Robustness and generalizability of our approach for post-hoc correction.
5. CR model behavior error analysis.

7.1 Post-Hoc OCR Error Correction

All post-hoc OCR correction approaches under comparison significantly reduce the amount of OCR errors. Tables 2 and 3 provide an overview of the performance as measured through WER and CER metrics.

eval–1. Table 2 shows the results for competing approaches for the *eval–1* scenario. This scenario mainly shows how well the models generalize in terms of language evolution, where instances come from books written in a different century. Note that, apart from the *temporal dimension*, another important aspect is that of *publisher's* specific attributes. Dependent on the publisher, there are orthographic variations, vocabulary, and other stylistic features, such as font-face, and so on.

In principle, low WER translates into fewer word segmentation (WS) errors, with WS errors being some of the most frequent errors (cf. Figure 5). Hence, reducing WER is critical for post-hoc OCR correction models. Our model, **CR**, achieves the best performance with the lowest score of WER=5.98%. This presents a relative decrease of $\Delta = 82\%$ compared to the WER in the original OCR text snippets. In terms of CER we have a relative decrease of $\Delta = 66\%$, namely, with CER=2.07%.

	WER	CER
OCR	33.3	6.1
CH	7.64 (▼77%)	2.79 (▼54%)
$CH_{\lambda=0.4}$	7.46 (▼78%)	2.53 (▼59%)
PB	11.45 (▼66%)	3.05 (▼50%)
Transformer	8.11 (▼76%)	2.24 (▼63%)
CR	**5.98** (▼82%)*	**2.07** (▼66%)*

Table 2: Correction results for *eval–1*. CR achieves *highly significant* (*) improvements over the best baseline CH_λ.

	WER	CER
OCR	32.3	5.4
CH	4.08 (▼87%)	1.32 (▼76%)
$CH_{\lambda=0.3}$	4.09 (▼87%)	1.35 (▼75%)
PB	9.21 (▼71%)	1.93 (▼64%)
Transformer	4.50 (▼86%)	**1.07** (▼80%)*
CR	**3.59** (▼89%)*	1.31 (▼76%)

Table 3: Correction results for *eval–2*. CR obtains *highly significant* (*) improvements over the best baseline CH_λ for WER, while Transformer has significantly the lowest CER.

Comparing our approach CR against CH_λ (the best competing approach in *eval–1*), we achieve highly significant ($p < .001$) lower WER and CER scores, as measured according to the non-parametric *Wilcoxon signed-rank* test with correction.[7] For WER and CER, CR compared to CH_λ obtains a relative error reduction of 21.7% and 25.8%, respectively. This shows that ConvNets allow for flexibility in capturing the different constituents of a word compound, that in turn may result in either *over* or *under* segmentation error.

Against the other competitors the reduction rates are even greater. Transformers has the lowest CER among the competitors, yet compared to CR its CER has a 8% relative increase. PB, performs the worst, mainly due to the character shifts (left or right) incurred due to word segmentation errors. Thus, strictly enforcing the attention mechanism along *very close* or the *same* positions

[7] We test for *normality* of distributions, and conclude that the produced WER and CER measures do not follow a normal distribution.

in the encoder-decoder results in sub-optimal post-hoc OCR correction behavior.

eval–2. Table 3 shows the results for the *eval–2* scenario. Due to the randomized instances for training and testing, the models have greater ability in correcting OCR errors. Contrary to *eval–1*, where the models were tested on instances coming from later centuries, in this scenario, the models do not suffer from *language evolution* aspects and other *book* specific characteristics. Therefore, this presents an easier evaluation scenario.

Here too the models show a similar behavior as for *eval–1*. The only difference in this case being that our approach CR does not achieve the best CER reduction rates. CR obtains highly significant lower ($p < .001$) WER rates than the Transformer. On the other hand, Transformer achieves the best CER rates among all competitors ($p < .001$). The significance tests are measured using the non-parametric Wilcoxon signed-rank test.

This presents an interesting observation, showing that Transformers are capable in learning all the complex cases of character errors. This behavior can be attributed to their capability in learning complex intra-input and input-output dependencies. However, in terms of WER, we see that a large reduction is achieved through ConvNets in CR, yielding the lowest WER rates, with a relative decrease of 89% in terms of WER. This conclusion can be achieved when we inspect CH_λ, which is the ablated CR model without ConvNet encoders.

7.2 Ablation Study

In the ablation study we analyze the impact of the varying components introduced in **CR**.

ConvNet Layers. The number of layers provides different levels of abstractions in encoding the OCR input. Table 4 shows CR's performance with varying number of layers trained using the *standard* cross-entropy loss. Increasing the number of layers for $k > 5$ does not yield performance improvements. We note that for the different evaluation scenarios, the number of necessary layers varies. For instance, in *eval–2* the number of optimal layers is 3. This can be attributed to the higher diversity of errors in the randomized validation instances, and thus, the need for more layers to capture the OCR errors.

Loss Function. The loss function in § 5.3 rewards higher the model's correcting behavior.

	eval–1		*eval–2*	
	WER	**CER**	**WER**	**CER**
$\text{CR}_{k=1}$	**6.18**	**2.15**	3.72	1.29
$\text{CR}_{k=2}$	6.46	2.30	4.18	1.46
$\text{CR}_{k=3}$	6.47	2.26	3.61	**1.26**
$\text{CR}_{k=4}$	6.93	2.51	**3.54**	1.31
$\text{CR}_{k=5}$	6.63	2.40	3.92	1.38
$\text{CR}_{k=6}$	6.68	2.52	3.94	1.50
$\text{CR}_{k=7}$	6.58	2.60	3.90	1.50
$\text{CR}_{k=8}$	6.56	2.48	3.64	1.54
$\text{CR}_{k=9}$	6.59	2.69	3.84	1.60
$\text{CR}_{k=10}$	6.32	2.52	3.61	1.62

Table 4: WER and CER values for **CR** with varying number of ConvNet layers trained using *standard loss* function.

	eval–1		*eval–2*	
	WER	**CER**	**WER**	**CER**
$\text{CR}_{\lambda=0.1}$	6.22	2.16	**3.59**	**1.31**
$\text{CR}_{\lambda=0.2}$	6.31	2.17	3.79	1.42
$\text{CR}_{\lambda=0.3}$	**5.98**	**2.07**	4.24	1.51
$\text{CR}_{\lambda=0.4}$	6.37	2.17	3.90	1.33
$\text{CR}_{\lambda=0.5}$	6.37	2.16	3.83	1.45
$\text{CR}_{\lambda=0.6}$	6.63	2.22	3.90	1.41

Table 5: WER and CER results for CR with different λ for *custom loss function*.

Table 5 shows the ablation results for CR with varying λ values for $\mathcal{L}'$ and fixed ConvNet layers ($k = 1$ and $k = 3$) as the best performing configurations in Table 4. Here too due to the different characteristics of the evaluation scenarios, different λ values are optimal for CR. We note that for *eval–1*, a higher λ of 0.3 yields the best performance. This shows that for diverging train and test sets (e.g., *eval–1*), the models need more stringent guidance in distinguishing copying from correcting behavior.

7.3 Page Level Performance

Evaluation results in § 7.1 convey the ability of models to correct erroneous input at snippet level. However, there are challenges on applying post-hoc correction models on real-world OCR transcriptions, which do not have their textual content separated into coherent and non-overlapping snippets.

action	*description*
S	accuracy of token segmentation
M	accuracy of token merging
R	accuracy of token character replacement (insertion/update/delete)

Table 6: Page level actions are used to measure the model's performance at page level.

page	**S**	**M**	**R**	*actions*
9	0.878 (66)	-	0.737 (19)	85
10	0.976 (83)	-	0.586 (29)	112
16	0.960 (50)	0.0 (1)	0.652 (23)	73
17	0.933 (90)	-	0.621 (29)	119

Table 7: Precision for **S**, **M**, **R** actions. In brackets are the number of undertaken actions, and the rightmost column has all actions.

In this section, for our model **CR**, at *page level* we assess the accuracy of undertaken actions in correcting the erroneous input text to its target form. Table 6 shows the set of actions that a model can undertake. We carry out a manual evaluation on an out-of-corpus book (book code `Z168355305`), that is not present in our ground-truth, for which we randomly sample a set of 4 pages.

We apply CR, namely, assess the accuracy of actions of correction during the decoding phase, over the OCR transcribed pages line by line with a window of 5 tokens. For each decoding step that produces an output that is *different* from the input, we assess the accuracy of that action. Table 7 shows the precision of CR for the different set of actions for the different pages. The results show that CR is robust and can be applied without much change even at page level with high accuracy of post-hoc correction behavior.

7.4 Robustness

We conduct a robustness test of the CR approach to check: (i) *in-group* post-hoc correction performance, where test instances come from the same books as the training ones, and (ii) *out-of-group*, where we train on one group and test on the rest of the groups. Table 8 shows the groups of books we use for (i) and (ii).

Table 9 shows the in-group and out-of-group post-hoc correction scores for CR when using

	#Train	#Dev	#Test	Book IDs
G1	312k	34.7k	86.1k	(8, 5, 12, 11)
G2	58.9k	6.5k	17.2k	(2, 1, 3, 10)
G3	217.3k	24k	59.8k	(4, 7, 6, 9)

Table 8: Book splits for assessing CR robustness.

	G1		***G2***		***G3***	
	WER	**CER**	**WER**	**CER**	**WER**	**CER**
OCR	28.1	5.7	34.0	7.1	31.2	5.8
standard loss function						
G1	10.1	2.9	24.7	6.3	18.9	4.9
G2	21.5	5.6	15.9	4.4	20.2	4.9
G3	16.9	4.4	18.9	4.8	10.4	2.6
custom loss function						
G1	10.1	2.8	24.2	5.6	18.4	4.4
G2	21.5	5.1	17.0	4.1	20.3	4.4
G3	16.9	4.3	18.7	4.6	10.3	2.5

Table 9: CR results with $k = 1$ trained using the standard and custom loss function with $\lambda = 0.1$.

a single ConvNet layer, using the standard and the custom loss functions, respectively. It can be seen that when the models are trained on a similar corpus (in-group), the error reduction is significantly higher compared to the evaluation on the out-of-group corpus. Furthermore, we note that the custom loss function consistently provides better trained models for post-hoc correction.

The results in Table 9 show that CR is robust providing highly significant decrease in terms of WER and CER, with an average of WER decrease of 52% for in-group with both the standard and custom loss. Whereas the out-of-group WER reduction is with 34% and 35% using the standard and custom loss, respectively. In terms of CER, for in-group we get a CER decrease of 47.6% and 50% for standard and custom loss, respectively. The advantage of the custom loss is shown for out-of-group evaluation, where the CER decrease is much more significant with 16.71% for standard loss function compared to 23.3% using the custom loss function.

From the three groups, when training on G3 the out-of-group post-hoc correction performance is the highest. This shows that on historical corpora, depending on the initial OCR error rate and possibly the error types due to the book's

characteristics impact significantly the correction performance.

7.5 Error Analysis

Here we analyze the structure of some typical errors that we fail to correct.

Word Segmentation. In terms of over-segmentation, the importance of the ConvNet layers in CR is shown when compared against CH and CH_λ. Common word segmentation *errors* for CH and CH_λ are, for example, *''Jndem''* to *''Jn''* and *''dem''*, *''Jedoch''* to *''Je''* and *''doch''*. *''vorbeyftreichen''* to *''vor beyftreichen''*. Most of these errors can be traced back to frequent constituents of the compound that exist in isolation too.

Character Error. There are easy character errors such as *''mein''* which is OCRed to *''mcin''* and is fixed by all approaches. However, for some words like *''löfcken''*, models like CH and Transformer correct them to the right word *''löfeten''*. CR fails to do so due to some frequent character bigrams such as *''ck''* that are very frequent in the dataset.

7.6 Dataset Limitations

The OCR quality can vary greatly across books, and from page to page. Based on manual inspection, we note that in some cases the WER can go well beyond 80%. It is expected that in such cases that the post-hoc OCR correction will vary too. Other possible issues include competing spellings for the same word, which may cause the models to encode conflicting information, yet, for transcribing historical texts, language normalization (i.e., opting for one spelling) is not recommended, as the meaning of the texts may change.

Language Evolution. There is a significant difference between *eval–1* and *eval–2* in terms of correction results. One explanation is due to the word spelling variations across centuries. Some examples include the substitution of single characters in words, which if not known would lead to systematic correction mistakes, e.g., $j \rightarrow i$, $v \rightarrow u$, $\text{ſ} \rightarrow s$, $\text{ä} \rightarrow \overset{e}{a}$. Accordingly, due to the missing information about the spelling change in *eval–1*, the corresponding WER and CER rates are higher.

8 Conclusion

In this work we assessed several approaches towards post-hoc correction. We find out that OCR transcription errors are contextual, and a large set are due word-segmentation, followed by word-errors. Models like Transformers have limited utility in this task, as pre-training is difficult to undertake, given the scarcity of historical corpora.

We proposed a OCR post-hoc correction approach for historical corpora, which provides flexible means to capturing various OCR transcription errors that are subject to *language evolution*, *typeface* and *book layout* issues. Through our approach CR we achieve great WER reduction rates with 82% and 89% for *eval–1* and *eval–2* scenarios, respectively.

Furthermore, ablation studies show that all the introduced components in CR yield consistent improvement over the competitors. Apart from post-hoc correction performance at snippet level, CR proved to be robust at page level too, where the undertaken correction steps are highly accurate.

Finally, we construct a release a new dataset for post-hoc correction of historical corpora in German language, consisting of more than 850k parallel textual snippets, which can help facilitate research for historical and low-resource corpora.

Acknowledgments

This work was partially funded by Travelogues (DFG: 398697847 and FWF: I 3795-G28).

References

Haithem Afli, Zhengwei Qiu, Andy Way, and Páraic Sheridan. 2016. Using SMT for OCR error correction of historical texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23–28, 2016*. European Language Resources Association (ELRA).

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing

by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17–21, 2015*, pages 349–359. The Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D15-1041

Adrien Barbaresi. 2016. Bootstrapped OCR error detection for a less-resourced language variant. In *Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016, Bochum, Germany, September 19–21, 2016*, volume 16 of *Bochumer Linguistische Arbeitsberichte*.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1–8, 2000*, pages 286–293. ACL. **DOI:** https://doi.org/10.3115/1075218.1075255

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics. **DOI:** https://doi.org/10.18653/v1/P16-1160

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*, pages 876–885. **DOI:** https://doi.org/10.18653/v1/N16-1102

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941. PMLR.

Rui Dong and David Smith. 2018. Multi-input attention for unsupervised OCR correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers*, pages 2363–2372. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P18-1220

Markus Dreyer, Jason Smith, and Jason Eisner. 2008. Latent-variable modeling of string transductions with finite-state methods. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25–27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1080–1089. ACL.

Noura Farra, Nadi Tomeh, Alla Rozovskaya, and Nizar Habash. 2014. Generalized character-level spelling error correction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 161–167. The Association for Computer Linguistics. **DOI:** https://doi.org/10.3115/v1/P14-2027

Jonas Gehring, Michael Auli, David Grangier, and Yann N. Dauphin. 2017a. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver,*

Canada, July 30 – August 4, Volume 1: Long Papers, pages 123–135. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P17-1012, **PMID:** 28964987, **PMCID:** PMC6754825

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017b. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. **DOI:** https://doi.org/10.1162/neco.1997.9.8.1735, **PMID:** 9377276

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18–21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1700–1709. ACL.

Yoon Kim, Yacine Jernite, David A. Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA*, pages 2741–2749. AAAI Press.

Yann LeCun, Yoshua Bengio, and others. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. 2015. Character-based neural machine translation. *CoRR*, abs/1511.04586.

William B. Lund, Douglas J. Kennard, and Eric K. Ringger. 2013. Combining multiple thresholding binarization values to improve OCR output. In *Document Recognition and Retrieval XX, part of the IS&T-SPIE Electronic Imaging Symposium, Burlingame, California, USA, February 5–7, 2013, Proceedings*, volume 8658 of *SPIE Proceedings*, page 86580R. SPIE.

William B. Lund, Eric K. Ringger, and Daniel David Walker. 2014. How well does multiple OCR error correction generalize? In *Document Recognition and Retrieval XXI, San Francisco, California, USA, February 5–6, 2014*, volume 9021 of *SPIE Proceedings*, pages 90210A–90210A–13. SPIE.

William B. Lund, Daniel David Walker, and Eric K. Ringger. 2011. Progressive alignment and discriminative error correction for multiple OCR engines. In *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18–21, 2011*, pages 764–768. IEEE Computer Society.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of Massive Datasets*, Cambridge University Press. 3.

Christian Reul, Uwe Springmann, Christoph Wick, and Frank Puppe. 2018a. Improving OCR accuracy on early printed books by utilizing cross fold training and voting. In *13th IAPR International Workshop on Document Analysis Systems, DAS 2018, Vienna, Austria, April 24–27, 2018*, pages 423–428.

Christian Reul, Uwe Springmann, Christoph Wick, and Frank Puppe. 2018b. State of the art optical character recognition of 19th century fraktur scripts using open source engines. *CoRR*, abs/1810.03436.

Gözde Gül Sahin and Mark Steedman. 2018. Character-level models versus morphology in semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers*, pages 386–396. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P18-1036, **PMID:** 30102173

Carsten Schnober, Steffen Eger, Erik-Lân Do Dinh, and Iryna Gurevych. 2016. Still not there? Comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December*

11–16, 2016, Osaka, Japan, pages 1703–1714. ACL.

Sarah Schulz and Jonas Kuhn. 2017. Multi-modular domain-tailored OCR post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017*, pages 2716–2726. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D17-1288

Miikka Silfverberg, Pekka Kauppinen, and Krister Lindén. 2016. Data-driven spelling correction using weighted finite-state methods. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, pages 51–59. Association for Computational Linguistics, Berlin, Germany. **DOI:** https://doi.org/10.18653/v1/W16-2406

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Jules Gagnon-Marchand and LJQ. n.d. https://github.com/Lsdefine/attention-is-all-you-need-keras/blob/master/transformer.py.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Ziqi Wang, Gu Xu, Hang Li, and Ming Zhang. 2014. A probabilistic approach to string transformation. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1063–1075. **DOI:** https://doi.org/10.1109/TKDE.2013.11

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *CoRR*, abs/1603.09727.

Shaobin Xu and David A. Smith. 2017. Retrieving and combining repeated passages to improve OCR. In *2017 ACM/IEEE Joint Conference on Digital Libraries, JCDL 2017, Toronto, ON, Canada, June 19–23, 2017*, pages 269–272. IEEE Computer Society.