Recognizing Learner Handwriting Retaining Orthographic Errors for Enabling Fine-Grained Error Feedback

Christian Gold[1], Ronja Laarmann-Quante[2] and Torsten Zesch[1]
[1]CATALPA, FernUniversität in Hagen, Germany,
[2]Ruhr University Bochum, Faculty of Philology, Department of Linguistics, Germany

Abstract

This paper addresses the problem of providing automatic feedback on orthographic errors in handwritten text. Despite the availability of automatic error detection systems, the practical problem of digitizing the handwriting remains. Current handwriting recognition (HWR) systems produce highly accurate transcriptions but normalize away the very errors that are essential for providing useful feedback, e.g. orthographic errors. Our contribution is twofold: First, we create a comprehensive dataset of handwritten text with transcripts retaining orthographic errors by transcribing 1,350 pages from the German learner dataset FD-LEX. Second, we train a simple HWR system on our dataset, allowing it to transcribe words with orthographic errors. Thereby, we evaluate the effect of different dictionaries on recognition output, highlighting the importance of addressing spelling errors in these dictionaries.

1 Introduction

Early L1 learners typically write by hand, even in the digital age, and handwriting remains important (Ray et al., 2022; Danna et al., 2022; Mathwin et al., 2022). Automatic feedback on error types in learner language is available (Laarmann-Quante, 2017; Berkling and Lavalley, 2015), but faces the practical problem of having to digitize the handwriting first. Current *handwriting recognition* (HWR) systems yield very good results (Kizilirmak and Yanikoglu, 2022; Xiao et al., 2020; Li et al., 2021) with one crucial problem: they typically normalize away the orthographic errors (Neto et al., 2020) that are important for giving useful feedback to learners. In Figure 1, when humans read this handwritten word, they look at the shapes of the letters to form hypotheses. The first letter(s) could be a *d* or a *cl* and we decide about this informed by a hypothesis about the whole word. In this case, we see that it is probably supposed to be *dounut*, so the first letter is a *d*. We see that there is an extra letter *u* at the third position which we ignore for forming our hypothesis about the word, but still recognize so that we could give a learner appropriate feedback about it.

Automatic handwriting recognition systems are typically trained and evaluated on handwritten text along with transcripts that do not contain orthographic errors. Many HWR systems contain a language model component (Scheidl et al., 2018) that is used to further normalize the output. As a result, HWR systems yield 'clean' transcripts without any orthographic errors (right branch in Figure 1) that cannot be used to give feedback on orthographic errors. Instead, we need HWR systems outputting transcripts that retain orthographic errors (middle branch in Figure 1).

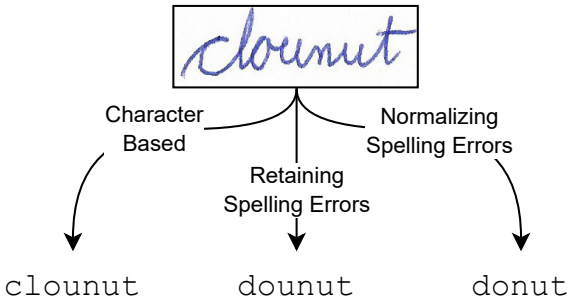

Figure 1: Handwritten example for different hypotheses (e.g. with and without normalizing spelling errors).

In this paper, we tackle this problem by first creating a dataset of handwritten text with transcripts retaining orthographic errors. For that purpose, we created comprehensive transcription guidelines (Gold et al., 2023) that precisely define our transcription goal. This is necessary as handwritten text contains other artifacts beyond orthographic errors, such as strikethroughs or inserts that we need to transcribe. In total, we transcribe 1,350 handwritten pages from German learners and thus create a dataset that is comparable in size to widely used English datasets like IAM (Marti and Bunke,

Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 352–360

2002) and CVL (Kleber et al., 2013).

Given this dataset, we are then able to quantify to what extent existing baseline systems are unable to transcribe handwritten text, especially if we only use the underlying character recognition probabilities. We compare this with training the HWR system on parts of our data, enabling it (in theory) to learn to correctly transcribe words with orthographic errors.

Furthermore, we change the dictionary used in the HWR system to also include systematic learner errors created by an automated generator. Note that providing the actual feedback is outside the scope of this paper. Here, we focus on analyzing the problem of turning an image of handwritten text into a digitized transcript, which is currently the main obstacle to applying existing feedback methods on a scale.

2 Existing Datasets

For training and evaluating a handwriting recognition system that retains orthographic errors, we need a dataset combining images of learner handwriting with transcripts containing orthographic errors. To our knowledge, no such dataset exists.

IAM and CVL are mostly in English and are often used to evaluate handwriting recognition systems. IAM in its version 3.0 is an extensive dataset and consists of about 1,500 pages with more than 13,000 text lines written by 650 adults, with different segmentation levels and corresponding transcripts. CVL is comparable to IAM with about 1,600 pages from 310 adult writers. The set consists of six English and one German text and thus has a slightly increased alphabet as the German Umlauts (ä, ö, and ü) are included. In comparison to IAM, it is only transcribed word-wise, ignoring most punctuation marks or strikethrough words, although a segmentation of text lines is available.

The Growth-In-Grammar GIG dataset (Durrant and Brenchley, 2018) is a learner dataset that retained orthographic errors. However, the corresponding image data is not available.

In contrast to GIG, FD-LEX (Becker-Mrotzek and Grabowski, 2018) is another learner dataset with published image data. In comparison to IAM and CVL where the participants copied a presented text by hand, this dataset consists of texts that were freely written based on a picture or a short story, and thus, more errors were made. Albeit, the transcripts from the FD-LEX dataset normalize orthographic errors and ignore other noise (e.g. strikethroughs).

Set	GYM_5	GYM_9	IGS_5	IGS_9	Sum
1	144	90	84	72	390
2	102	96	84	108	390
3	132	138	114	60	444
4	120	138	90	90	438
5	156	132	72	84	444
6	162	120	96	114	492
7	168	144	132	120	564
8	150	132	120	120	522
9	138	144	126	114	522
10	138	144	132	132	546
11	150	120	108	90	468
12	144	84	108	72	408
Test Set		91		Total:	5628
Annotator 1		168			
Annotator 2		1092			

Table 1: Statistics of the complete FD-LEX Dataset and our transcription effort. Cells in green are subsets for the test set; dark orange and blue are transcribed by Annotator 1 and Annotator 2, respectively.

In conclusion, none of the existing datasets fulfills our need for available image data and a transcript containing orthographic errors.

3 Dataset Creation

As no suitable dataset is available, we need to build one. We decided to use the German learner corpus FD-LEX as a starting point, as it already contains scans of learner handwriting with a sufficient number of orthographic errors. Looking at the example in Figure 2, we can see additional typical challenges for automatic handwriting recognition e.g. strikethroughs and inserts.

FD-LEX was built as a corpus for analyzing the writing competence of learners. It covers two different German school types: *Gymnasium* (GYM) ('academic track school') and *Integrierte Gesamtschule* (IGS) ('comprehensive school') from two grades (5^{th} and 9^{th}) each. It has about 5,600 scanned color pages from about 940 children and is thus exceeding the IAM (1,500 pages) and CVL (1,600 pages) datasets in size. A detailed listing can be seen in Table 1. As stated, the transcript provided with the corpus was created under another focus (e.g. normalizing orthographic errors), thus we had to transcribe it anew.

3.1 Transcription Guidelines

We first created transcription guidelines (Gold et al., 2023) to formulate rules on how to deal with different situations while creating an authentic transcrip-

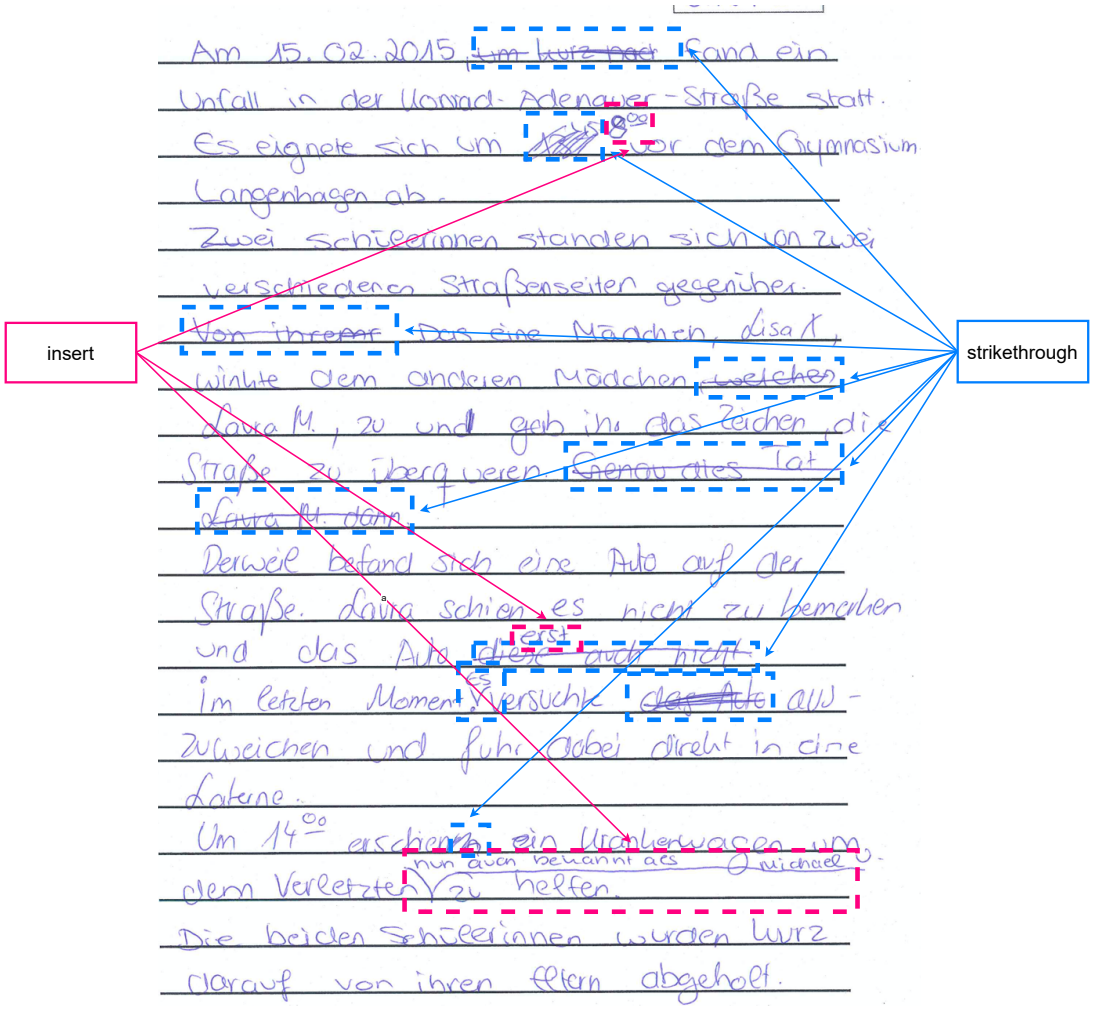

Figure 2: An example of FD-LEX with common transcription challenges like strikethroughs and inserts.

tion of the written form.[1] Following the guidelines should yield an exact transcript of the handwritten forms while at the same time allowing conversion into readable text automatically. This approach ensures that the transcribed text accurately reflects the writing skills of the learner and enables researchers to identify any patterns or issues related to spelling deficiencies.

We now describe the main issues covered in the guidelines:

Text/line alignment One line of text in the image must correspond to the line of text in the transcript.

Content Only the handwritten content of the learner should be transcribed. This excludes the printed text of the paper sheet as well as drawn figures.

Indistinct characters must be placed within curly brackets {}. When in doubt between two characters, the transcription should reflect the character that is appropriate in the given context. Learners may attempt to deceive teachers when uncertain whether a word should begin with a capital letter[2] or not, resulting in both versions being written on top of each other. In such cases, both letters should be enclosed in curly brackets and separated by a plus (+) sign, with the first letter in curly brackets being the correct one in the context.

Spacing should be carefully analyzed and considered in the context of the individual writing style. In cases where a gap between characters of the same word is noticeably larger than the average space between words, the spacing should be transcribed within curly brackets to indicate the deviation from the norm: {S }chool.

[1]The transcription guidelines can be found at https://github.com/catalpa-cl/learner-handwriting-recognition.

[2]Particularly, since nouns are capitalized in German.

Spelling errors are transcribed exactly as they appear in the original text, without any correction or modification.

Strikethrough characters, words, lines When a character or a word is struck through, the transcript should represent the number of characters with a hash sign (#). If a line is made invalid in the same manner, the line is transcribed with three hash signs (###).

Inserts Direct inserts should be transcribed enclosed in curly brackets with a less-than sign, like `< text`. Indirect inserts, which are written at a different location such as at the end of a page, can be indicated by an asterisk (*) and a number if there are multiple inserts. These indirect inserts should be transcribed where they appear in the image. To do this, an `{insert1 *}` tag is added in the line where the text should be inserted, and the actual insert content is transcribed at the location where it appears with: `{insert1 text}`.

Punctuation marks, special characters, emoticons All punctuation marks have to be transcribed as they appear, with the only exception that they should align with grammar rules in regard to spacing: correct: (`However,`) incorrect: (`However␣,`). Special characters are treated individually for e.g. tally marks[3] are transcribed with an ampersand (&) `{|&}`.

While using special signs and encoding (e.g. at inserts or tally marks, strikethroughs), a conversion between different target transcriptions can be achieved, e.g. a) for a line-wise transcript of the genuine content to be used for HWR; or b) for a coherent text where inserts are inserted and the text-line alignment is broken up to be used for semantic analysis.

3.2 Annotation Process

Following the guidelines, we re-transcribed about 1,250 pages, each by one annotator. To diversify our dataset, we transcribed the first 3 sets of each school type and grade (colored cells of Table 1). To assess the quality of the transcripts, some pages were transcribed by both annotators and the inter-annotator agreement (IAA) was computed. The double-annotation was done repeatedly during the whole transcription period and differences between the transcripts were discussed among annotators.

[3]To keep track of word counts, the learners use vertical strokes after every ten words. We refer to them as tally marks.

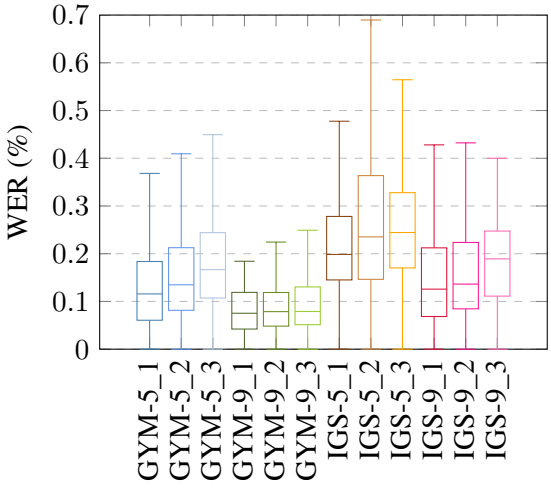

Figure 3: Distribution of Word Error Rates (WER) between the original FD-LEX dataset transcription and our error-retraining transcription.

In this way, a total of about 90 pages (subparts in green, see Table 1) were transcribed in parallel and both transcripts were merged into a gold transcription by an adjudicator. We achieved an IAA between both annotators of .98 on the character level and an IAA of .99 between both annotators and the gold label.[4]

3.3 Dataset Analysis

Transcribing the data allowed us to examine the distribution of orthographic errors, i.e. spelling, word separation, and capitalization. For that purpose, we aligned our new transcripts with the original transcripts using word alignment and measured the *word error rate* (WER). As strikethroughs are words that were made invalid, they would only increase WER and thus were excluded from our analysis.

In Figure 3, it can be observed that there are many differences between our transcripts and the original transcripts, suggesting that the use of the original transcripts may not be ideal for HWR. Additionally, the results in Figure 3 show that the 9th grade had fewer errors compared to the 5th grade, while the GYM performed better than the IGS for both grades.

4 Baseline Experiments

To track our recognition performance improvements, we create a baseline by training a straightfor-

[4]While some characters may appear unclear to one annotator and the other annotator may see it differently, we decided to calculate the IAA by ignoring curly brackets.

ward handwriting recognizer on our dataset. Commonly, the performance of the recognizer is evaluated with two metrics, namely *character error rate* (CER) and *word error rate* (WER). While CER gives numerical feedback on how many characters have been misread by the recognizer, WER measures how many words are different from the gold-standard transcription. This means that lower values indicate better recognition performance. For the purpose of this paper's focus on word-level analysis, we will concentrate on WER rather than CER.

4.1 Recognizer Setup

For our experiments, we use a recognizer based on a *convolutional neural network* (CNN) architecture combined with a *connectionist temporal classification* (CTC) (Graves et al., 2006) for decoding. The designed architecture reduces the text-line images from 2048x128 to 128x96 (Time-steps x Charset) in 7 CNN-layers, 2 BLSTMs, and a final dense layer. This architecture is based on Scheidl (2018), with CTC decoding and additional *word beam search* (WBS) for language-model decoding (Scheidl et al., 2018)[5]. We extended the character set used in the recognizer from 80 to 95 characters to cover all German Umlauts ('Ä', 'Ö', 'Ü', 'ä', 'ö', 'ü') and 'ß' as well as additional punctuation marks and special characters like '€'.

We use a text-line level recognizer and thus need a text-line segmentation. Thus, we first reduced the colored scans to gray level and removed ruled lines as proposed by Gold and Zesch (2022). To segment the full pages into text-lines we use a segmentation with the A^* path finding algorithm. This algorithm works on a binary image and tries to find a path through the text lines while avoiding crossing handwritten strokes.

4.2 Baseline Setup

To train the recognizer we first used as much data as possible and combined IAM (∼11,300 lines) and CVL (∼13,400 lines) with our dataset (∼12,200 lines). Furthermore, we use the gold transcripts which were transcribed by both annotators. These 91 pages (see Table 1) contain about 1,000 text-lines and are referred to as test set in the following. With the described setup and the combined training data, the recognition performance results in a CER of 11.5% and a WER of 37.6% on our test set.

As our dataset matches IAM and CVL in size, we decided to train the recognizer again based on our dataset only (without IAM and CVL). With this setup, we were able to improve the recognition performance slightly with a CER of 10.7% and a WER of 34.7% on our test set. With these recognition results, we decided to use this setup as our Baseline (Table 2).

5 Decoding with Dictionary Constraint

Most research and publicly available databases for HWR pertain to adults. In these cases, spelling errors are typically ignored because they are estimated to be rare and not important to be kept in the output. Therefore, the predicted words can be mapped to a large dictionary of possible words, which has been shown to yield better recognition rates, as recognition errors can be eliminated this way (Scheidl et al., 2018).

5.1 Path Decoding and Word Beam Search

The standard method to map the Neural Network (NN) results to a text string is the CTC (Graves et al., 2006). In a more detailed manner, the NN returns a matrix containing the probability distribution for each character along so-called time-steps along the line of text. The matrix is then further analyzed by a beam search decoder such as the vanilla beam search by Hwang and Sung (2016).

However, without deeper knowledge, the beam search algorithm could randomly output an indistinguishably written character like 'a' as 'o', if the probability is the same. To avoid this, a commonly employed approach involves constraining the generated output to words that are contained in a predefined dictionary. This can be done with WBS as introduced by Scheidl et al. (2018).[6] However, with traditional dictionaries which only contain correctly spelled words, spelling errors would be eliminated from the texts.

5.2 Lower Bound

The ideal dictionary would consist of the vocabulary of the learners as well as the orthographic variants. To find out what the performance would be with such an ideal dictionary, i.e. to determine the lower bound for WER that would be possible with such a dictionary, we compiled a dictionary

[5] https://github.com/githubharald/SimpleHTR, https://github.com/githubharald/CTCWordBeamSearch

[6] Although the proposed algorithm of WBS includes a more sophisticated language model, we did not make use of it as the dictionary is increased enormously and thus increases the computational costs.

from our transcripts of the test set. This means that this dictionary only contains words that appear in the texts to be recognized as well as the specific orthographic variants that are present in the texts.

Using this dictionary in the WBS decoder, we can reduce the WER from 34.7% to 25.0%. Compared to the baseline, this is an improvement of the WER of 10 percentage points, i.e. almost one-third. With the ideal dictionary, further recognition improvements could only be achieved by changing the model or training data. This means, that the achieved performance can be seen as the Lower Bound that we want to approach.

5.3 German Learner Dictionary

For our purpose, we need a German dictionary covering the vocabulary of young learners in the first place. We decide to use childLex (Schroeder et al., 2015) for this purpose.[7] The childLex corpus was created by extracting word forms from over 500 children's books with a target age between 6 and 12 years. Although this age range does not cover the 9th-grade students from our dataset, it seems better suitable than a dictionary compiled from adult language. To slightly restrict the extensive vocabulary, we use a subset that comprises all word forms that occurred in at least ten different books (an arbitrary cutoff point)[8]. This is supposed to exclude rare and specialized words, which could distract the recognizer from choosing words that are generally much more likely to appear in a text. In total, the dictionary compiled this way contains about 45,000 word forms.

Using this dictionary in Word Beam Search, i.e. constraining the output possibilities to the dictionary words, resulted in a WER of 29.6%, which is an improvement of 5 percentage points compared to the baseline, see Table 2, row 'WBS childLex'.

5.4 Specific Dictionary

Since childLex is a generic dictionary compiled from books, it does not cover the whole vocabulary of the FD-LEX dataset. Therefore, we compiled another dictionary from the original transcripts of the FD-LEX dataset (in which orthographic errors were normalized) with a total of ∼11,850 words. Although the dictionary is smaller than the one compiled from childLex, it benefits from containing only words which the learners wrote in relation to the topics of the dataset. For example, one of the texts is about an accident with a cyclist and therefore, 20 compound words containing the German word for 'bicycle' appear in the dictionary, whereas only 9 such words appear in the childLex dictionary. Overall, there is an overlap of about 7,150 words between the FD-LEX dictionary and the childLex dictionary.

	CER	WER
Baseline	10.7	34.7
WBS childLex	11.3	29.6
WBS childLex + SP	10.0	30.1
WBS FD-LEX	12.4	31.3
WBS FD-LEX + SP	9.9	29.0
WBS childLex + FD-LEX + SP	9.0	25.9
Lower Bound	10.8	25.0

Table 2: Results obtained with and without using the WBS, and using different dictionaries. SP indicates dictionaries that are expanded to include spelling errors.

Incorporating the FD-LEX dictionary instead yielded a notable improvement in recognition performance at the word level compared to the baseline, achieving a WER of 31.3%, see Table 2, row 'WBS FD-LEX'. However, it fell slightly short of the recognition accuracy obtained with the childLex dictionary.

6 Spelling Error Generator

To approximate the Lower Bound (see Section 5.2), spelling variants must be added to the dictionary. Thus, we generate possible (systematic) spelling errors based on the procedure described in Laarmann-Quante (2016). We generate possible misspellings for all words in the childLex and FD-LEX dictionaries. The error generation procedure works as follows: A correctly spelled word is automatically enriched with linguistic information such as phonemes, syllables, and morphemes, based on the web service G2P of the Bavarian Archive of Speech Signals (BAS) (Reichel, 2012; Reichel and Kisler, 2014)[9], see also Laarmann-Quante et al. (2019a) for more information about these annotations. The information is then used to analyze (via a set of rules) which systematic errors could be made on this word. By systematic we mean that particular

[7] For the English community we want to mention a similar corpus https://www.sketchengine.eu/oxford-childrens-corpus/.

[8] More precisely, if a word form is included, all related word forms with the same lemma are included as well.

[9] https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/Grapheme2Phoneme/

principles of German orthography are violated, e.g. consonant doubling (**komen* for *kommen*, eng.: 'to come')[10], a syllabic principle, or final devoicing (**Walt* for *Wald*, eng.: 'forest'), a morphological principle (see Eisenberg, 2006 for the theoretical framework). We also generate errors reflecting the overuse of such principles, e.g. **Walld* for *Wald*. Errors that cannot be explained via such principles (such as a seemingly random omission of a letter as in **Wad* for *Wald*) are not generated because there is an infinite number of ways in which a word could be misspelled. We assume, however, that using the systematic errors in the sense described above, should capture most of the errors that the pupils commit because they are the major obstacles when learning how to spell in German.

In total, 57 different error categories can be generated (not all apply to each word, though, while some words may contain multiple instances of the same error category, e.g. when there are two doubled consonants in one word such as *Wasserfall*, eng.: 'waterfall'). The error categories that can be generated can be found in Laarmann-Quante et al. (2019b).[11]

Of course, more than one error can be committed within a word. We account for this by including all possible combinations of up to 2 systematic errors that apply to a word. Including *all* possible error combinations would lead to an exponential increase of misspellings to consider, most of which will be highly unlikely, though.

6.1 Coverage of the Dictionaries

Applying the spelling error generation to all words in a dictionary results in an enormous increase in the number of word forms. As shown in Table 3, for the childLex dictionary, the number of words rises from 45,000 (row 2) to about 14 million (row 3). Likewise, FD-LEX with 11,000 words (row 4) rises to 3.6 million words (row 5).

As we see in the last column of the table, the original dictionaries only cover 74% (childLex) or 88% (FD-LEX) of the word forms present in the test set. Including the generated spelling errors, the coverage increases by 7-8 percentage points. However, even if FD-LEX and childLex and the spelling errors are combined (row 6 in Table 3), not all word forms are covered (90%).

[10] We mark misspellings with an asterisk (*) in this paper.

[11] Under the levels PGI and PGII ('Phoneme-Grapheme Correspondence Level'), SL ('Syllabic Level'), and MO ('Morphematic Level')

Dictionary	# Words	Coverage
test set	1,472	100
childLex	45,347	74
childLex + SP	13,993,376	82
FD-LEX	11,874	81
FD-LEX + SP	3,670,962	88
FD-LEX + childLex + SP	15,990,735	90
FD-LEX + childLex + SP + Case	-	94

Table 3: Number of words and coverage of the test set vocabulary (in percent) for various dictionary settings. *SP* indicates dictionaries with added spelling errors, *Case* indicates that letter case variants are considered.

A manual inspection showed that one reason that not all vocabulary was covered, is that words may be capitalized at sentence beginnings in the texts, but the dictionaries do not contain capitalized variants of all words. However, including upper- and lowercase variants for all words would nearly double the size of the vocabulary, which is computationally not feasible for WBS. However, it shall be mentioned that the inclusion of both letter cases increases the coverage rate to approximately 94% (row 7 in Table 3).

We further investigated the last 6% of missing coverage, which is 88 words. 30 of these were caused by incorrect word separation (14 words that were incorrectly written together; 9 interrupted words due to line-breaks; 5 separated words due to strict transcription (e.g. huge gap after the first character); and 2 miscellaneous cases). Another 24 words were not covered due to a missing letter and 3 times two letters were swapped. These are 'unsystematic' errors that were not generated. For 19 words, the errors were not covered by the generator but they appeared systematic in a sense that one may think of further rules to generate them in the future, e.g. if 'i' follows 'l' the learner tends to write 'di' instead of 'li'. The few words left were not covered for various reasons, e.g. interference with transcription rules, more than 2 errors in the word, and 2 non-words (number plate of a car).

6.2 Influence of the Advanced Dictionaries

In the following, we include the dictionaries (with and without generated spelling errors) in the decoding process of the HWR system with WBS to see if the recognition performance can be improved.

The results are shown in Table 2. We see in rows 'WBS childLex' and 'WBS FD-LEX' that including a dictionary (without spelling errors) already improves the recognition performance compared

to the Baseline by 3-4 percentage points in terms of WER.

However, adding spelling errors into the dictionary did not necessarily improve the performance. For childLex, the WER increases by 0.5 percentage points when spelling errors are added to the dictionary (compare rows 2 and 3). As discussed in Section 6.1, by adding spelling errors, the number of word forms included in the dictionary is increased extremely. Hence, chances are high that a wrong spelling variant or a spelling variant of another word is chosen. In contrast, the FD-LEX dictionary is more restricted to the vocabulary of the learners and thus could benefit from adding spelling variants: The recognition performance is increased by 1 percentage point when compared to the dictionary without spelling errors (see rows 4 and 5).

The best result was achieved by combining both dictionaries and their spelling errors. This way, the WER decreases to 25.9% and is thus within 1 percentage point of the Lower Bound.

7 Conclusion and Further Work

In this paper we tackled the issue of retaining orthographic errors when automatically recognizing learner handwriting. This is a prerequisite for giving automated feedback on spelling performance based on handwritten texts.

We created a handwriting recognition dataset of German learner texts based on the FD-LEX dataset by transcribing 1,350 pages using new transcription guidelines. The utilization of a dictionary to restrict the output resulted in an improvement of our baseline. Furthermore, our results indicate that incorporating generated spelling errors leads to an improvement in recognition performance at the word level, with the error rate decreasing from 35% to 25%, representing a decrease of 10 percentage points.

Although we were able to cover 94% of the originally used words using a spelling error generator, the huge number of words in the dictionary raises questions about its practicality. Therefore, one of the next goals should be to allow more probable errors while avoiding overwhelming the dictionary. Therefore, further analysis is necessary to determine which errors were made by learners in FD-LEX and which ones were addressed by the generated errors. This information can be used to reduce the size of the error set by eliminating unnecessary or rare errors. Additionally, an analysis of common error combinations can aid in generating more targeted errors while avoiding redundant ones.

Furthermore, the focus of this study was not on improving the recognition model itself. However, recognition improvements could be made by implementing a more sophisticated model like full page recognition as introduced by Bluche et al. (2017).

Acknowledgments

This work was partially conducted at "CATALPA - Center of Advanced Technology for Assisted Learning and Predictive Analytics" of the FernUniversität in Hagen, Germany.

References

Michael Becker-Mrotzek and Joachim Grabowski. 2018. FD-LEX (Forschungsdatenbank Lernertexte). Textkorpus Scriptoria. Köln: Mercator-Institut für Sprachförderung und Deutsch als Zweitsprache. Available at: https://fd-lex.uni-koeln.de, DOI: 10.18716/FD-LEX/861.

Kay Berkling and Rémi Lavalley. 2015. WISE: A Web-Interface for Spelling Error Recognition for German: A Description and Evaluation of the Underlying Algorithm. In *GSCL*, pages 87–96.

Théodore Bluche, Jérôome Louradour, and Ronaldo Messina. 2017. Scan, Attend and Read: End-to-end Handwritten Paragraph Recognition with MDLSTM Attention. In *14th International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1050–1055. IEEE.

Jérémy Danna, Marieke Longcamp, Ladislas Nalborczyk, Jean-Luc Velay, Claire Commengé, and Marianne Jover. 2022. Interaction between Orthographic and Graphomotor Constraints in Learning to Write. *Learning and Instruction*, 80:101622.

P. Durrant and M. Brenchley. 2018. Growth in Grammar Corpus.

Peter Eisenberg. 2006. *Das Wort*, 3rd edition, volume 1 of *Grundriss der deutschen Grammatik*. J.B. Metzler, Stuttgart.

Christian Gold, Ronja Laarmann-Quante, and Torsten Zesch. 2023. Preserving the Authenticity of Handwritten Learner Language: Annotation Guidelines for Creating Transcripts Retaining Orthographic Features. In *1st Computation and Written Language (CAWL) Workshop at ACL*.

Christian Gold and Torsten Zesch. 2022. CNN-Based Ruled Line Removal in Handwritten Documents. In *18th International Conference on Frontiers of Handwriting Recognition (ICFHR)*, pages 530–544. Springer.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Kyuyeon Hwang and Wonyong Sung. 2016. Character-Level Incremental Speech Recognition with Recurrent Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5335–5339. IEEE.

Firat Kizilirmak and Berrin Yanikoglu. 2022. CNN-BiLSTM model for English Handwriting Recognition: Comprehensive Evaluation on the IAM Dataset.

Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. 2013. CVL-Database: An Off-line Database for Writer Retrieval, Writer Identification and Word Spotting. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 560–564. IEEE.

Ronja Laarmann-Quante. 2016. Automating Multi-Level Annotations of Orthographic Properties of German Words and Children's Spelling errors. In *Proceedings of the 2nd Language Teaching, Learning and Technology Workshop (LTLT)*, pages 14–22.

Ronja Laarmann-Quante. 2017. Towards a Tool for Automatic Spelling Error Analysis and Feedback Generation for Freely Written German Texts Produced by Primary School Children. In *7th International Workshop on Speech and Language Technology in Education (SLaTE)*, pages 36–41.

Ronja Laarmann-Quante, Stefanie Dipper, and Eva Belke. 2019a. The Making of the Litkey Corpus, a Richly Annotated Longitudinal Corpus of German Texts Written by Primary School Children. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 43–55, Florence, Italy. Association for Computational Linguistics.

Ronja Laarmann-Quante, Anna Ehlert, Katrin Ortmann, Doreen Scholz, Carina Betken, Lukas Knichel, Simon Masloch, and Stefanie Dipper. 2019b. The Litkey Spelling Error Annotation Scheme: Guidelines for the Annotation of Orthographic Errors in German Texts. *Bochumer Linguistische Arbeitsberichte (BLA)*.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2021. TrOCR: Transformer-Based Optical Character Recognition with Pre-Trained Models. *arXiv preprint arXiv:2109.10282*.

U-V Marti and Horst Bunke. 2002. The IAM-Database: An English Sentence Database for Offline Handwriting Recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 5(1):39–46.

Kathryn P Mathwin, Christine Chapparo, and Joanne Hinnit. 2022. Children with Handwriting Difficulties: Developing Orthographic Knowledge of Alphabet-Letters to Improve Capacity to Write Alphabet Symbols. *Reading and Writing*, 35(4):919–942.

Arthur Flor de Sousa Neto, Byron Leite Dantas Bezerra, and Alejandro Héctor Toselli. 2020. Towards the Natural Language Processing as Spelling Correction for Offline Handwritten Text Recognition Systems. *Applied Sciences*, 10(21):7711.

Karen Ray, Kerry Dally, Leah Rowlandson, Kit Iong Tam, and Alison E Lane. 2022. The Relationship of Handwriting Ability and Literacy in Kindergarten: A Systematic Review. *Reading and Writing*, pages 1–37.

Uwe D. Reichel. 2012. PermA and Balloon: Tools for String Alignment and Text Processing. In *INTERSPEECH*.

Uwe D. Reichel and Thomas Kisler. 2014. Language-Independent Grapheme-Phoneme Conversion and Word Stress Assignment as a Web Service. In R. Hoffmann, editor, *Elektronische Sprachverarbeitung: Studientexte zur Sprachkommunikation 71*, pages 42–49. TUDpress.

Harald Scheidl. 2018. Build a Handwritten Text Recognition System Using TensorFlow - A Minimalistic Neural Network Implementation which can be Trained on the CPU.

Harald Scheidl, Stefan Fiel, and Robert Sablatnig. 2018. Word Beam Search: A Connectionist Temporal Classification Decoding Algorithm. In *International Conference on Frontiers of Handwriting Recognition (ICFHR)*, pages 253–258. IEEE.

Sascha Schroeder, Kay-Michael Würzner, Julian Heister, Alexander Geyken, and Reinhold Kliegl. 2015. childLex: A Lexical Database of German Read by Children. *Behavior Research Methods*, 47:1085–1094.

Shanyu Xiao, Liangrui Peng, Ruijie Yan, and Shengjin Wang. 2020. Deep Network with Pixel-Level Rectification and Robust Training for Handwriting Recognition. *SN Computer Science*, 1:1–13.