Evaluating Automatic Spelling Correction Tools on German Primary School Children's Misspellings

Ronja Laarmann-Quante
Department of Linguistics
Faculty of Philology
Ruhr University Bochum, Germany

Lisa Prepens and **Torsten Zesch**
CATALPA - Center of Advanced Technology
for Assisted Learning and Predictive Analytics
FernUniversität in Hagen, Germany

Abstract

Most existing spellcheckers have been developed for adults and it is yet understudied how well children's texts can be automatically spellchecked, e.g. to build tools that assist them in spelling acquisition. This paper presents a detailed evaluation of six tools for automatic spelling correction on texts produced by German primary school children between grades 2 and 4. We find that popular off-the-shelf tools only achieve a correction accuracy of up to 46 % even when local word context is taken into account. For many misspellings, the desired correction is not even among the suggested candidates. A noisy-channel model that we trained on similar errors, in contrast, achieves a correction accuracy of up to 69 %. Further analyses show that this approach is very successful at candidate generation and that a better re-ranking of correction candidates could lead to a correction accuracy of ~90 %. Most of the remaining misspellings are so distorted that they are hard to correct without broader context. Furthermore, we analyze how the tools perform at different grade levels and for misspellings with different edit distances.

1 Introduction

Assisting children in learning to spell correctly is a time-consuming task and it requires solid diagnostic skills in order to tell different kinds of spelling errors apart. For example, misspelling the German word *Hund* ('dog') as **Hunt* includes an error of final devoicing, which does not change the word's pronunciation (we mark incorrect spellings with an asterisk). In contrast, misspelling the word as **Hunb* comprises a mirrored letter in the first place and the pronunciation is affected. Thus, the different kinds of errors require different feedback or different kinds of practice exercises for the child.

Therefore, automated tools for spelling error classification have been proposed (Berkling and Lavalley, 2015; Laarmann-Quante, 2017). However, when children are free to write whatever they want, in contrast to dictations, it is a non-trivial task to find out which words they wanted to write before the spelling errors can be analyzed. In the above example, the popular spellchecking tool Hunspell[1] would correct **Hunt* to *Hund* but **Hunb* to *Hub* '((vertical) lift)', leading to a wrong analysis of the child's errors. Hence, before the types of errors can be analyzed, misspellings first have to be detected and corrected. In the following, we will concentrate on the automatic correction step.

The aim and contribution of this paper is an evaluation of six existing spelling correction tools on misspellings of German primary school children taken from the Litkey Corpus (Laarmann-Quante et al., 2019). We examine how well the existing approaches perform in order to be used e.g. in an automatic spelling error diagnosis tool. Thereby, we set a baseline for future approaches tailored towards German children's spellings. Furthermore, we analyze the spelling correction performance of the tools for errors with different edit distances and for different grade levels. We assume that over time, children's errors get more adult-like, leading to a better performance of the tools.

The remainder of this paper is structured as follows: Section 2 introduces related work about the evaluation of spellcheckers and approaches for the correction of children's errors. In Section 3, we introduce the Litkey Corpus, which is used as the data basis for our spelling correction experiments. The experimental setup for the evaluation study is explained in Section 4, and Section 5 presents the results including some further analyses.[2]

2 Related Work

Spelling correction tools have mostly been compared on English data and often artificial errors

[1] http://hunspell.github.io

[2] Data and experimental code from this study are available under https://github.com/catalpa-cl/spellchecker-evaluation-german-children.

(see e.g. Näther, 2020). However, it is well-known that conventional spellcheckers are tailored towards errors produced by proficient adults (e.g. typos) and struggle with errors containing multiple edits, as e.g. produced by language learners (Rimrott and Heift, 2008; Flor et al., 2019). Bexte et al. (2022) introduced a multilingual benchmark data set of spelling errors produced by language learners in order to compare spellcheckers on. They found that for the Litkey data, correction performance was poorer compared to data of Italian children and data of second-language learners of German, indicating that spelling errors of German children are rather hard to correct. However, they only compared three spellchecking tools (Hunspell, LanguageTool and DKPro-Spelling, which was introduced in their paper) and used an uncleaned version of the Litkey Corpus. In the corpus, some proper names occur so frequently that they could potentially bias the correction performance as they do not appear in the spellcheckers' dictionaries. In the study we present in this paper, we will do some data cleaning in order to reduce corpus-specific artifacts and compare six spellcheckers on the data, some of them trained on similar errors. Furthermore, we will provide a more in-depth analysis of the tools' performances across different grade levels and for errors with different edit distances.

While several spellchecking approaches that target errors of foreign language learners have been proposed (e.g. Boyd, 2009; Hovermale, 2011; Flor and Futagi, 2012; Nagata et al., 2017), children's errors have rarely been addressed. Downs et al. (2020, 2022) present a spelling correction approach for English children based on phonetic similarity, which outperforms existing spellcheckers. For German, a similar approach was taken by Stüker et al. (2011). However, they found that the phonetic model alone could not outperform Hunspell. Therefore, in our study, we focus on the performance of existing spellchecking tools to set a baseline for future approaches that target German children's errors.

3 Data Set

We base our study on the Litkey Corpus (Laarmann-Quante et al., 2019), which is a freely-available longitudinal corpus consisting of 1,922 German texts written by 251 primary school children between the second half of grade 2 and the end of grade 4 (= end of primary school). Every few months, at ten testing points in total, the same children were asked to write down a story that was shown in a sequence of six pictures. At the end of each school year, i.e. at the second, sixth and tenth testing point, the same picture story was used, all other picture stories were different.

The corpus includes the manual transcription of the handwritten texts (which we refer to as *orig* in the following), as well as a target hypothesis for each word with a manual correction of orthographic errors (referred to as *target*). Note that the target hypothesis does not correct grammatical or other kinds of errors.

3.1 Data Cleaning

The original data set consists of 212,505 orig-target pairs (6,364 target types). For our experiments, we removed the following kinds of tokens:

- (target) tokens with less than 2 alphabetic characters in order to only capture words and not punctuation marks or artifacts like *(grade) 4b*
- words that are marked in the corpus as non-identifiable, non-existing/non-standard or as containing illegible characters
- the proper names *Lea*, *Lars* and *Dodo* because they are specific to the corpus and appear multiple times in every text so they would distort the statistics
- words that contain a dot (capturing abbreviations)

Furthermore, we removed all special annotation marks from the remaining tokens, e.g. linebreak markers. This leaves us with 162,426 orig-target pairs in total.

3.2 Misspelling Statistics

In the present study, we are not looking at pure capitalization errors because they are a special type of error which require knowledge of sentence structure and morphosyntax (in German, the head of a noun phrase is capitalized). Therefore, in this paper, we do not count tokens as misspellings if orig and target only differ with regard to letter case. We also ignore wrong word separations, e.g. when the child wrote **aufeinmal* for *auf einmal* ('suddenly') or **zu frieden* for *zufrieden* ('pleased'). This leaves us with a total of 24,601 misspellings.

On average, the (cleaned) texts consist of 84 (± 40) words with an average misspelling rate of

	total	misspelled
# orig-target pairs	162,426	24,601
# unique orig-target pairs	15,188	9,484
# unique target words	5,675	3,154

Table 1: Basic statistics of the cleaned data set.

17 % (± 11). Only 9 texts contain no misspelling at all. Some further statistics about the cleaned data set and the number of misspellings are shown in Table 1.

For some target words, we find many different spelling variants. The top 3 are *Fundbüro* 'lost-and-found office' (68 variants), *glücklich* 'happy' (56 variants) and *Karton* 'cardboard box' (51 variants).

A particular challenge for automatic spellchecking are origs that have to be corrected to different targets, depending on the context. For example, *bas* is corrected to *dass* 'that', *Bus* 'bus' and *pass* 'pay (attention)', respectively, in the gold standard correction. In our data set, we find 935 such origs (1,855 if also different letter case is taken into account). This number also includes real-word errors such as *als* 'as', which is found as a correct word but also as a misspelling of *alles* 'all' in the corpus.

3.3 Development over Testing Points

Due to the longitudinal design of the Litkey Corpus, it is possible to analyze the development of misspellings over a time period of 2.5 years. Table 2 shows some statistics for each of the ten testing points in the cleaned Litkey data set. Since the number of available texts per testing point differs, some testing points contribute more errors to the whole data set (in absolute numbers) than others. Furthermore, we see that over time, the children produce longer texts but that the error rates per text decrease.

We hypothesize that the children's increasing spelling competence is not only reflected by a decrease in error rate but also that the errors become more adult-like so that they are easier to correct by conventional spellchecking systems. As discussed in Section 2, spellcheckers typically struggle with misspellings that have a high edit distance to the target word. Figure 1 shows the proportion of errors with a particular edit distance per testing point. We use an edit distance where deletions, insertions and substitutions each have a cost of 1. We see a clear trend that over time, misspellings with an edit distance > 1 become rarer. There is some oscillation, which may be due to the fact that different picture stories elicited very different words but if we compare testing points 02, 06 and 10, where the same picture story was used, we find a steady decrease of higher edit distances. Nevertheless, it is noteworthy that already in grade 2, more than two thirds of the errors only have an edit distance of 1. Recall that pure capitalization errors are not part of our misspelling data set so that the prevalence of an edit distance of 1 that we see here is not attributable to words that only differ in letter case.

testing point	grade	∅ err. rate/text	∅ # errors/text	∅ text length	# errors abs.	# texts abs.
01	2	.29	16	54	1,716	141
02	2	.26	17	68	2,154	165
03	3	.26	17	67	2,028	162
04	3	.25	21	86	2,520	173
05	3	.22	20	92	2,527	173
06	3	.18	19	104	2,924	231
07	4	.18	18	105	2,900	223
08	4	.18	22	124	3,046	215
09	4	.13	17	126	2,549	217
10	4	.12	15	120	2,237	222

Table 2: Basic statistics for each testing point.

4 Experimental Setup

Spelling correction is typically seen as a two-step process, consisting of misspelling detection and misspelling correction (see e.g. Hládek et al., 2020). The misspelling detection step usually relies on a dictionary lookup and its performance is largely dependent on the coverage of the dictionary. Bexte et al. (2022) achieved an F-Score of up to .79 for error *detection* in German primary school children's texts from the Litkey Corpus, which is higher than the results for most second-language learner corpora that were investigated in that study. Hence, in this paper, we only concentrate on the correction step, which has been shown to be much more problematic for children's texts. That is, we use the gold standard set of misspellings as the basis for our experiments.

4.1 Spellcheckers

While the number of existing spellchecking approaches is abundant (see e.g. Hládek et al., 2020), we restrict our comparison to six correction systems available for German. Four of them are usable off-the-shelf and the other two have to be trained based on a list of misspellings and their correc-

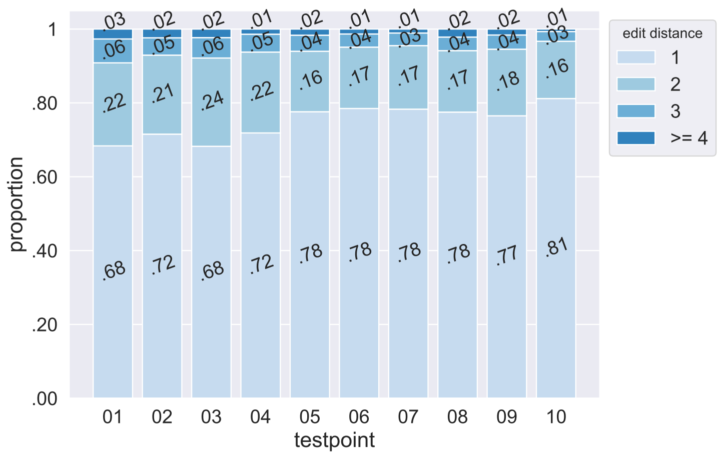


Figure 1: Distribution of edit distances between orig and target for each testing point.

tions. We exclude neural approaches, which are not readily available for German and would require more training data. For those spellcheckers which allow to specify a full-form dictionary (which are the two trainable ones and DKPro-Spelling), we try two different ones, namely **Hun-dict**, which is the Hunspell dictionary converted into a full-form word list that was also used for the correction experiments reported in Bexte et al. (2022) and **childLex** (Schroeder et al., 2015), which is compiled from 500 German children's books.

4.1.1 Off-the-Shelf Spellcheckers

We use all off-the-shelf spellcheckers with default configurations unless noted otherwise.

Hunspell is one of the most popular spellchecking libraries and used e.g. in OpenOffice and macOS. It finds correction candidates by different means, e.g. by applying edit operations to the misspelled string or by computing the similarity with words in the dictionary.[3] For our experiments, we use Hunspell with the German dictionary it comes with and simply feed all misspelling types into the system, as there is no context awareness.

Nuspell[4] is similar to Hunspell and can be used with the same dictionaries. Like Hunspell, it supports rich morphology and complex word compounding, which is important for German.

LanguageTool[5] is an open-source proofreading tool with add-ons for several popular programs like MS Word or Google Docs. It has a built-in dictionary (based on Hunspell with extensions) and mainly relies on handcrafted rules, which are partly context-sensitive. Therefore, we use LanguageTool in two configurations: firstly, we only feed individual misspellings into the tool, i.e. we ignore the context, and secondly we spellcheck the words in the context of the whole text to benefit from context-sensitive rules. Note that in this case we do not clean the texts as rigorously as described in Section 3.1. We only remove special annotation marks (e.g. linebreak markers) as well as words that are marked as non-identifiable or as non-existing word forms in order to maintain the necessary context information. For spellchecking whole texts with LanguageTool, we first disabled two internal rules, i.e. capitalization at the beginning of a sentence (UPPERCASE_SENTENCE_START) and spaces before/behind commas and brackets (COMMA_PARENTHESIS_WHITESPACE). The reason is that these rules would always fire first and prevent the search for a proper correction candidate. For example, a misspelled word at the beginning of a sentence would only be corrected to uppercase although it is still misspelled (e.g. **dan* → **Dan* rather than *Dann* 'then').

DKPro-Spelling[6] (Bexte et al., 2022) is a spellchecking toolkit that can be integrated into an NLP processing pipeline in the DKPro framework (Eckart de Castilho and Gurevych, 2014). It is highly customizable but also comes with a preconfigured setting, which we use for our experiments. In this setting, three correction candidates are chosen from a dictionary based on the smallest edit distance on the character level. Note that in the case of ties, DKPro-Spelling returns more than

[3] See https://zverok.space/spellchecker.html for details.
[4] https://nuspell.github.io
[5] https://github.com/languagetool-org/languagetool
[6] https://github.com/catalpa-cl/ltl-spelling

three candidates. In a second step, the candidates are re-ranked based on a Web1T trigram language model (Brants, 2006).

4.1.2 Trainable Spellcheckers

We train the two trainable spellcheckers in our experiment on a total of 7,488 unique misspellings (case-sensitive) and their manual corrections from two other German spelling corpora that consist of children's texts. These are the H1 corpus (Berkling, 2016), which includes texts from second and third grade and the Osnabrücker Bildergeschichtenkorpus ('Osnabrück picture story corpus'; Thelen, 2000, 2010), which mainly contains texts of children in second grade. While we ignore pure capitalization errors in our data set, for the remaining errors we will include a case-sensitive evaluation (see Section 4.3), which is why we keep letter case information in the training data.

Brill & Moore We use a Java implementation of the noisy channel approach presented in Brill and Moore (2000)[7]. The model learns the probability of certain edits from the training data, which can also comprise several characters at once. For example, from orig-target pairs like **faren* - *fahren* ('to drive'), **Fart* - *Fahrt* ('drive'), the model may learn that instead of the sequence *ahr*, children often write *ar*. That is, it uses contextual information in that it does not only learn that *h* is often omitted but also in which context. Thus, the model is able to learn specific error patterns of children that are present in the training data. Note that the model is only context-sensitive in the sense that it can take into account the context of an edit on the character level but not the broader context of the surrounding words. The tool outputs a fixed number of 10 correction candidates per default and we leave it like that.

Norma[8] (Bollmann, 2012) was originally developed for spelling normalization of historical language data but can be used on all kinds of non-standard language. It is a toolchain that combines different normalization techniques. We use the default setting in which first, whole word forms are mapped to one another. If no mapping is applicable, context-sensitive character rewrite rules are applied, and third, if no rules are applicable, the correction is chosen based on weighted Levenshtein distance by choosing the word from the dictionary with the lowest distance. All steps are learnt from training data. Note that Norma always only outputs the one most probable correction candidate.

[7] https://github.com/adrianeboyd/BrillMooreSpellChecker

[8] https://github.com/comphist/norma

4.2 Upper Bound

For most texts, spellcheckers are not able to achieve 100 % correction accuracy simply because some of the target words are not part of the underlying dictionary and hence cannot be suggested as correction candidates (e.g. certain proper names or rare compounds). We therefore compute the upper bound for the performance of each spellchecker in our experiments.

For Hunspell and LanguageTool, we determine the upper bound by feeding the target words into the respective tool. If no suggestion is made, the word is recognized as correct, i. e. the target word is contained in the dictionary. In order to find the upper bound when letter case is ignored, we capitalize all target words, since e.g. verbs and adjectives are recognized as correct even if they are capitalized, but nouns are recognized as false if they are lowercased.

For the other spellcheckers, the upper bound can be determined directly by checking how many of the target words are contained in Hun-dict and childLex, respectively. Note that DKPro-Spelling uses an adapted version of childLex where only words that occur in at least ten children's books are considered (45k types) whereas Brill & Moore and Norma use the full childLex word list (158k types), which results in slightly different upper bounds.

4.3 Evaluation Setup

We measure the correction performance of a spellchecker in two ways: We evaluate a) how often the target word is ranked at the first rank of the suggestion list of the respective spellchecker (FIRST) and b) how often the target word is contained somewhere in the suggestion list (ALL). We suppose that all spellcheckers provide an internal ranking, so that the most probable candidate is ranked first, although it is often not made explicit (except for DKPro-Spelling and Brill & Moore). Hence, the FIRST metric, which we also call correction accuracy, is relevant for fully automatic spelling correction and therefore the one we are most interested in here. The ALL metric is not directly comparable across spellcheckers because they produce different numbers of suggestions (see Table 3). However, it tells us how often a spellchecker does in principle generate the right correction candidate.

	avg. # suggs		
Hunspell	4.8	±	4.3
Nuspell	5.3	±	4.9
LangTool (words)	15.2	±	8.4
LangTool (texts)	15.2	±	7.9
DKPro (childlex)	9.0	±	8.7
DKPro (hun-dict)	10.1	±	10.0
Brill & Moore	10.0	±	0.0
Norma	1.0	±	0.0

Table 3: Average number of suggestions per spellchecker, including cases with 0 suggestions.

Pushing the candidate to the first rank could then be achieved via a second step where the candidates are re-ranked e.g. based on the context.

We furthermore distinguish between an evaluation based on types versus tokens as well as a case-sensitive and a case-insensitive evaluation, resulting in four different conditions, see Table 4. By the distinction of tokens vs. types, we mean that when we evaluate the spellcheckers based on tokens, we count every single occurrence of a misspelling and whether it was corrected successfully or not. When we look at types, we count the correction of every unique misspelling (= unique orig-target pair) only once.[9] Hence, when spellcheckers do not take the context into account and correct the same misspelling always in the same way, the evaluation on a token base can be strongly influenced by misspellings that occur very frequently. The evaluation on a type base, in contrast, shows more clearly the correction performance on different errors. If a spellchecker performs better on tokens than on types, it means that (some) misspellings with a high frequency are corrected more successfully than low-frequency misspellings and vice versa.

Capitalization is highly context-dependent (see Section 3.2). Therefore, the performance of a spellchecker may be underestimated when letter case is taken into account. We are mostly interested in how often a spellchecker is able to suggest the correct word, irrespective of lettercase. Nevertheless, we also report the case-sensitive results in order to see what large a role capitalization plays for a successful automatic correction.

[9]Note that, for example, **Hunt* - *Hund* and **Hunb* - *Hund* are two different types although they share the same target word. Likewise, *alls* - *als* ('when') and *alls* - *alles* ('all') share the same orig word but we treat them as two different types.

5 Results

Table 4 shows the performance of each spellchecker based on the FIRST and ALL metric and the upper bound (UB) for each of the four evaluation conditions (types vs. tokens, case-sensitive vs. case-insensitive). The upper part of the table contains the off-the-shelf spellchecking tools and the lower part the trained spellcheckers. DKPro-Spelling is special in that it is the only off-the-shelf spellchecker in which the default configuration comes with a re-ranking of candidates based on local context. Therefore, the same misspelling may be corrected differently based on context, hence there is no type-based evaluation for this spellchecker. The respective dictionary is indicated in brackets. Recall that for LanguageTool, we tried two configurations, a) based on a list of errors (*LangTool words*) and b) based on the errors within context (*LangTool texts*). Hence, there is again no type-based evaluation for the latter configuration. Since Norma only outputs one correction candidate, the results for the FIRST and ALL metric are identical. Therefore, we only list them under FIRST.

Among the **off-the-shelf spellcheckers**, DKPro-Spelling has the best performance. Regarding the FIRST metric, this may be due to the context-based re-ranking of candidates. Therefore, we will reconsider the other spellcheckers with language model re-ranking in Section 5.1. Among Hunspell, Nuspell and LanguageTool, differences are not large. None of them is able to rank the correct candidate on first rank in more than 40 % of cases. We see that this rather poor result is not primarily due to a bad ranking of suggestion candidates. Even when all correction candidates are considered, the correct one is only available for 62-71 % of all tokens. This means that even with a better re-ranking, the spellcheckers would not be able to correct every third to fourth word appropriately because the right correction is not even considered. Note that LanguageTool achieves slightly better results when only an error list is provided rather than the errors in context, which can be explained by more rules firing in the latter case that lead to an inappropriate correction.

The **trained spellcheckers** largely outperform the off-the-shelf tools in all conditions. Norma has a correction accuracy of up to 62 %. This shows that even without knowledge of the context, quite a good correction accuracy can be achieved when children's error patterns are taken into account. We

	Token (case ins.)			Token (case sens.)			Type (case ins.)			Type (case sens.)		
	FIRST	ALL	UB	FIRST	ALL	UB	FIRST	ALL	UB	FIRST	ALL	UB
Hunspell	.34	.62	(.94)	.33	.59	(.94)	.35	.56	(.93)	.32	.51	(.92)
Nuspell	.35	.64	(.94)	.34	.61	(.94)	.36	.59	(.93)	.34	.54	(.92)
LangTool words	.39	.71	(.98)	.37	.68	(.97)	.38	.72	(.99)	.35	.66	(.96)
LangTool texts	.37	.68	-	.36	.65	-	-	-	-	-	-	-
DKPro (chL)	.46	.80	(.92)	.44	.77	(.91)	-	-	-	-	-	-
DKPro (Hun)	.45	.76	(.93)	.44	.73	(.89)	-	-	-	-	-	-
Brill&Moore (chL)	.57	.92	(.97)	.53	.91	(.97)	.58	.84	(.95)	.52	.83	(.94)
Brill&Moore (Hun)	.53	.86	(.93)	.51	.83	(.89)	.48	.77	(.90)	.45	.73	(.86)
Norma (chL)	.62	-	(.97)	.56	-	(.97)	.54	-	(.95)	.50	-	(.94)
Norma (Hun)	.57	-	(.93)	.52	-	(.89)	.49	-	(.90)	.45	-	(.86)

Table 4: Overall evaluation results based on the FIRST and ALL metric for each of the four evaluation conditions (types vs. tokens, case-sensitive vs. case-insensitive). The dictionary used (childLex or Hun-dict) and the upper bound (UB) for each spellchecker are given in brackets.

see that the error patterns in two other German corpora of children's texts that were used for training generalize well enough to achieve good correction results also on the Litkey corpus. Most remarkably, for the Brill & Moore spellchecker, the desired correction is among the 10 correction candidates in >90 % of cases on the token level. Hence, a successful automatic correction is mainly a matter of candidate ranking here.

Some general observations can be made across all spellcheckers: The difference between **case-sensitive** and **case-insensitive** evaluation is rather small, indicating that proper capitalization is only a minor issue with regard to spelling correction in the children's texts.

With regard to **type-based** versus **token-based** evaluation, we see only small differences for most spellcheckers with a slight tendency towards better results on a token base. This indicates that the more frequently occurring misspellings are easier to correct than the rare ones. The difference is most pronounced for Norma, which may be explainable due to the fact that this tool stores particular correction patterns.

Finally, we can observe that the **upper bound** is mostly $>.90$ up to .99, which shows a very good coverage of the underlying dictionaries. For spellcheckers that we used with different dictionaries, we find that generally, childLex outperforms Hun-dict, i.e. a more child-directed dictionary is useful. For the following analyses, we therefore only use the results based on childLex for these spellcheckers.

5.1 Language Model Re-Ranking

A re-ranking of correction candidates based on local word context has been shown to be beneficial (Bexte et al., 2022). Therefore, we add a re-ranking to all spellchecker outputs based on the trigram model built from voxforge.org speech data that comes with the CMU Sphinx toolkit[10]. Unlike the Web1T model used by DKPro, this model is freely available and we suppose that speech data are close to the language that primary school children use in their writing. We try different conditions: re-ranking a) all candidates, b) only the top 5 and c) only the top 3 candidates. Table 5 shows for each condition, how often the desired correction ended up on the first rank.

For comparison, the first column shows the result without re-ranking or with default re-ranking in the case of DKPro-Spelling. Recall that for Norma, no re-ranking can be done since only one candidate is given. For DKPro-Spelling, although it comes with re-ranking off-the-shelf, we re-rank the candidates again with the CMU Sphinx language model for comparability. Note that DKPro-Spelling only outputs three correction candidates by default but it can be more if there are ties prior to re-ranking. We consider all these candidates for re-ranking, which is why we only report our new re-ranking results under "all candidates".

We see that for all spellcheckers, our re-ranking is beneficial, except for LanguageTool and DKPro-Spelling, where the default ranking works better. The best re-ranking results are achieved when

[10]https://cmusphinx.github.io/wiki/download/

	def.	3 cand.	5 cand.	all cand.
Hunspell	.34	.40	.40	.39
Nuspell	.35	.42	.41	.41
LTool words	.39	.36	.33	.24
DKPro	.46	-	-	.42
Brill & Moore	.57	.69	.64	.51
Norma	.62	-	-	-

Table 5: Correction accuracy (=first suggestion) after language model re-ranking (case insensitive). For comparison, the column *def.* ('default') repeats the first column of Table 4, i.e. the performance without re-ranking or default re-ranking in the case of DKPro.

only the top 3 candidates are re-ranked, indicating that the spellcheckers' original ranking (without knowledge of the context) is already quite useful in that lower-ranked candidates introduce more noise. Among the off-the-shelf spellcheckers, the improvements are only moderate, though, and none of them outperforms DKPro-Spelling. This means, the top result of a spellchecker that can be used off-the-shelf is a correction accuracy of 46 %, which means that fully automatic spelling correction would get more than every second word wrong.

For Brill & Moore, re-ranking the top 3 candidates improves the result by 12 percentage points, thereby outperforming Norma. Hence, the best result that could be achieved overall in this study is a correction accuracy of 69 % (we checked that only re-ranking the top 2 candidates did not improve the results any further). Given that in over 90 % of cases the desired correction is among the top 10 candidates for this spellchecker, there is still room for improving the re-ranking in future work to achieve a very high correction accuracy with this approach.

5.2 Comparison by Edit Distance

As stated in Section 2, common spellcheckers typically struggle with higher edit distances. For the trained spellcheckers in this study, we hypothesize that this is not so much the case because they can learn correction patterns that comprise several edits. To analyze the performance of the spellcheckers for different edit distances, we look separately at all misspellings with a particular edit distance and note how often the desired correction is at the first rank or another rank within the top 3 candidates (after re-ranking). The results for Brill & Moore and DKPro-Spelling (representing the best trained spellchecker and the best off-the-shelf spellchecker) are shown in Figure 2, the results for the other spellcheckers can be found in Appendix A. For DKPro-Spelling we use the default re-ranking here because it performed better than our re-ranking.

We see that for an edit distance of 1, all spellcheckers are able to find the desired correction for the majority of misspellings. However, even for the lowest edit distance, the trained spellcheckers Brill & Moore and Norma outperform the off-the-shelf spellcheckers, which shows that learning error patterns is even beneficial for seemingly easy errors. All spellcheckers have in common that the higher the edit distance, the less likely they are to provide the right correction. However, this is most pronounced for the off-the-shelf spellcheckers. For misspellings with an edit distance ≥ 4, off-the-shelf spellcheckers only correct a tiny fraction of words correctly, whereas Brill & Moore still includes the correct candidate in half of the cases.

5.3 Spellchecking Performance over Time

We saw earlier (Figure 1) that over the time course of primary school, the edit distances of the misspellings get smaller, which is why we expect the spellcheckers to work better on later testing points than on earlier testing points.

To analyze this, we look at the top 3 candidates (after re-ranking) at each of the ten testing points individually and note how often the desired correction is at the first rank or at one of the other ranks. Figure 3 shows the results for Brill & Moore and DKPro-Spelling (the latter again with default re-ranking). The results for all other spellcheckers are given in Appendix B.

We see that for Brill & Moore, the testing point does not have a big influence, which could be explained by the fact that the edit distance does not have such a big impact on this spellchecker (as was shown in Figure 2). In contrast, for the off-the-shelf spellcheckers such as DKPro-Spelling, where we saw a larger impact of edit distance, we can observe the expected trend that later testing points are easier to correct than early ones. However, we also find a lot of oscillation between testing points and in total, the differences are not very large. So even by the end of primary school, correction performance remains rather poor compared to the trained spellcheckers.

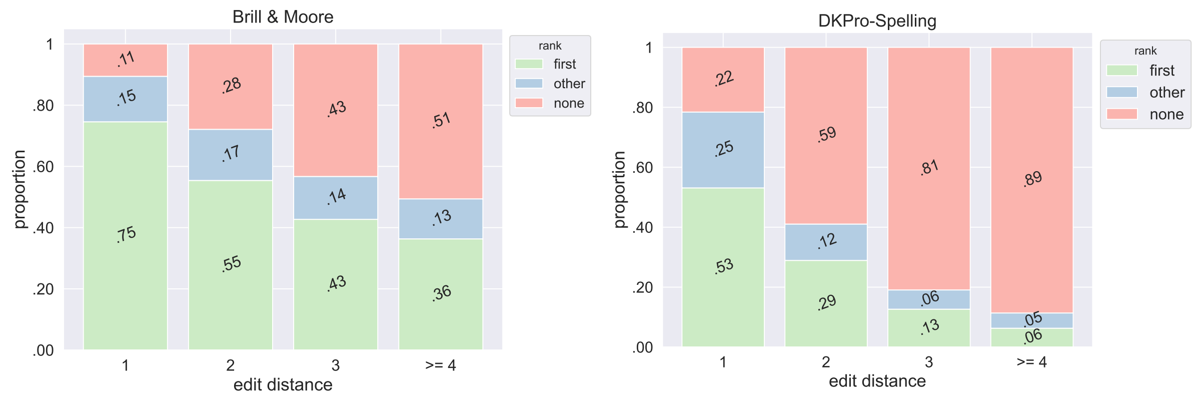


Figure 2: Correction performance per **edit distance** for Brill & Moore (left) and DKPro-Spelling (right). The coloring of the bars indicates from bottom to top how often the desired correction is ranked at the first rank, another rank or not among the suggestions when the top 3 candidates after re-ranking are considered (case-insensitive).

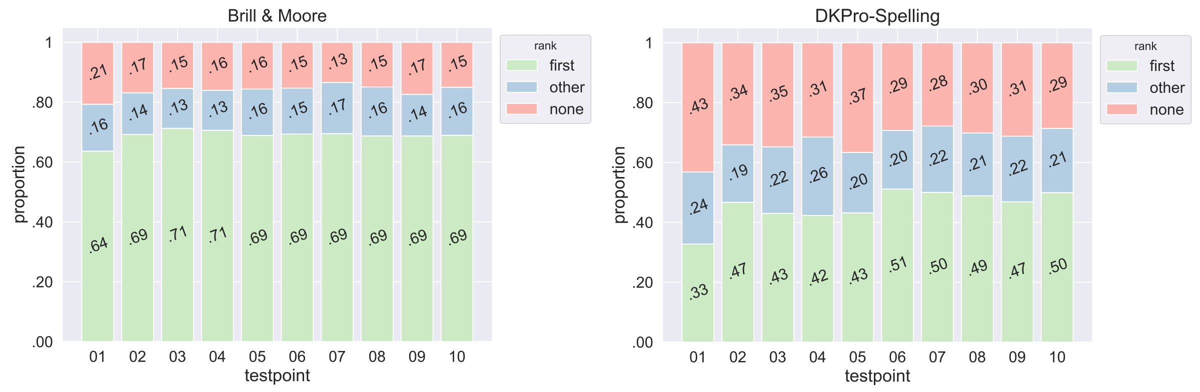


Figure 3: Correction performance per **testing point** for Brill & Moore (left) and DKPro-Spelling (right). The coloring of the bars indicates from bottom to top how often the desired correction is ranked at the first rank, another rank or not among the suggestions when the top 3 candidates after re-ranking are considered (case-insensitive).

5.4 Failure Analysis

In the following, we explore for what kinds of misspellings even the best spellchecking approach in this study (the **Brill & Moore** implementation) failed to find the right target word. We saw that on the (case-insensitive) type level, the Brill & Moore spellchecker had the right correction among the top 10 candidates in 84 % of cases given the childLex dictionary, with an upper bound of 95 % achievable corrections. In total numbers, this means that for 981 misspelling types (10.3 %), the target word was not among the top 10 candidates, although it would have been findable in the dictionary.

A deeper analysis shows that 23 % of these cases are real-word errors, i.e. the misspelling itself is contained in childLex. The remaining misspellings that could not be corrected have very high edit distances between orig and target, namely 2.4 on average. If we take the length of the target word into account (since an edit distance of 2 is more severe for short words than for long words), we find that on average, 46 % of the characters in the non-correctable words are wrong. Hence, the words are so distorted that without having broader context information in the first place, finding the right target word is almost impossible, even for humans. Some examples are given in Table 6.

orig	target	dist.	transl.
gawen	kaufen	3	‘buy’
niegs	nichts	3	‘nothing’
feid	fällt	4	‘falls’
glugeis	glücklich	6	‘happy’
sagras	zerkratzt	7	‘scratched’

Table 6: Examples of highly distorted words that the Brill & Moore spellchecker was not able to correct.

6 Conclusion

We compared six different spelling correction tools on German primary school children’s texts. We found very different performance behaviors between off-the-shelf tools on the one hand and tools that are trained on texts from other primary school children on the other hand.

We could see that off-the-shelf spellcheckers perform rather poorly across all grade levels. They only reach an overall correction accuracy of up to 46 % (including a trigram-model based re-ranking of candidates), which is certainly insufficient in order to be used e.g. in an automatic spelling error diagnosis tool. Furthermore, we saw that these spellcheckers are often not able to include the right correction in their suggestion lists at all, so that a better re-ranking would not help much.

Spellcheckers that learn error patterns from other German children's corpora are more successful in correcting, leading to an overall correction accuracy of up to 69 %. Most notably, the noisy-channel approach turned out to be very successful in candidate generation: in up to 92 % of cases, the target word was among the top 10 candidates. This means that there is the potential for future work to improve the fully automatic correction by finding more effective means of re-ranking the candidates. With regard to the remaining misspellings, we saw that they often include real-word errors and very distorted words, which could potentially be tackled by neural approaches where more context is taken into account but which also need more training data.

Acknowledgments

This work was partially conducted in the framework of CATALPA – Center of Advanced Technology for Assisted Learning and Predictive Analytics of the FernUniversität in Hagen, Germany. We would like to thank Marie Bexte for her support with DKPro-Spelling, Katrin Ortmann for her contributions in compiling the training data and the anonymous reviewers for their helpful comments.

References

Kay Berkling. 2016. Corpus for children's writing with enhanced output for specific spelling patterns (2nd and 3rd grade). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3200–3206.

Kay Berkling and Rémi Lavalley. 2015. WISE: A Web-Interface for Spelling Error Recognition Description and Evaluation of the Algorithm for German. In *International Conference of the German Society for Computational Linguistics and Language Technology*, GSCL, pages 87–96.

Marie Bexte, Ronja Laarmann-Quante, Andrea Horbach, and Torsten Zesch. 2022. Lespell - a multilingual benchmark corpus of spelling errors to develop spellchecking methods for learner language. In *Proceedings of the Language Resources and Evaluation Conference*, pages 697–706, Marseille, France. European Language Resources Association.

Marcel Bollmann. 2012. (Semi-)automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, Lisbon, Portugal.

Adriane Boyd. 2009. Pronunciation modeling in spelling correction for writers of English as a Foreign Language. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 31–36, Boulder, Colorado. Association for Computational Linguistics.

Thorsten Brants. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium.

Eric Brill and Robert C Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Brody Downs, Oghenemaro Anuyah, Aprajita Shukla, Jerry Alan Fails, Sole Pera, Katherine Wright, and Casey Kennington. 2020. KidSpell: A child-oriented, rule-based, phonetic spellchecker. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6937–6946, Marseille, France. European Language Resources Association.

Brody Downs, Maria Soledad Pera, Katherine Landau Wright, Casey Kennington, and Jerry Alan Fails. 2022. KidSpell: Making a difference in spellchecking for children. *International Journal of Child-Computer Interaction*, 32:100373.

Michael Flor, Michael Fried, and Alla Rozovskaya. 2019. A benchmark corpus of English misspellings and a minimally-supervised model for spelling correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 76–86, Florence, Italy. Association for Computational Linguistics.

Michael Flor and Yoko Futagi. 2012. On using context for automatic correction of non-word misspellings in student essays. In *Proceedings of the seventh* *workshop on building educational applications Using NLP*, pages 105–115.

Daniel Hládek, Ján Staš, and Matúš Pleva. 2020. Survey of automatic spelling correction. *Electronics*, 9(10):1670.

D. J. Hovermale. 2011. *Erron: A Phrase-Based Machine Translation Approach to Customized Spelling Correction*. Ph.D. thesis, Ohio State University.

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 *Proceedings of the Seventh ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, pages 36–41.

Ronja Laarmann-Quante, Katrin Ortmann, Anna Ehlert, Simon Masloch, Doreen Scholz, Eva Belke, and Stefanie Dipper. 2019. The Litkey Corpus: A richly annotated longitudinal corpus of German texts written by primary school children. *Behavior Research Methods*, 51(4):1889–1918.

Ryo Nagata, Hiroya Takamura, and Graham Neubig. 2017. Adaptive spelling error correction models for learner English. *Procedia Computer Science*, 112:474–483.

Markus Näther. 2020. An in-depth comparison of 14 spelling correction tools on a common benchmark. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1849–1857, Marseille, France. European Language Resources Association.

Anne Rimrott and Trude Heift. 2008. Evaluating automatic detection of misspellings in german. *Language Learning & Technology*, 12(3):73–92.

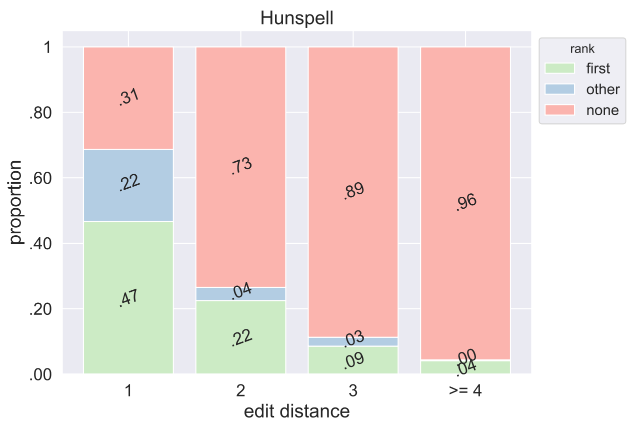
Sascha Schroeder, Kay-Michael Würzner, Julian Heister, Alexander Geyken, and Reinhold Kliegl. 2015. childLex: A lexical database of German read by children. *Behavior Research Methods*, 47(4):1085–1094.

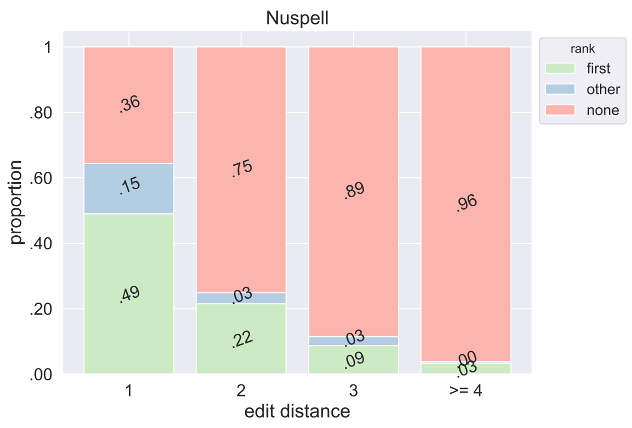
Sebastian Stüker, Johanna Fay, and Kay Berkling. 2011. Towards context-dependent phonetic spelling error correction in children's freely composed text for diagnostic and pedagogical purposes. In *Twelfth Annual Conference of the International Speech Communication Association*.

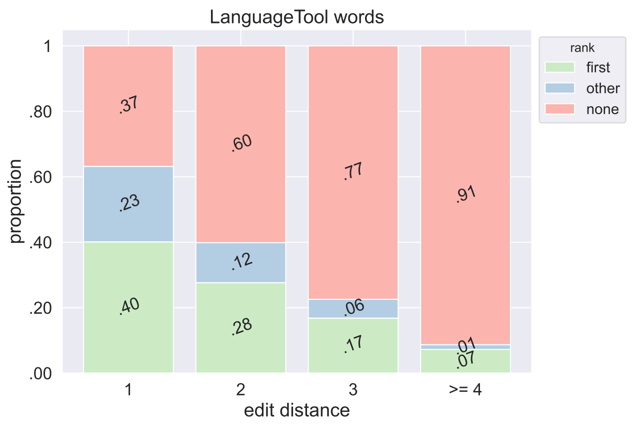
Tobias Thelen. 2000. Osnabrücker Bildergeschichtenkorpus: Version 1.0.0.

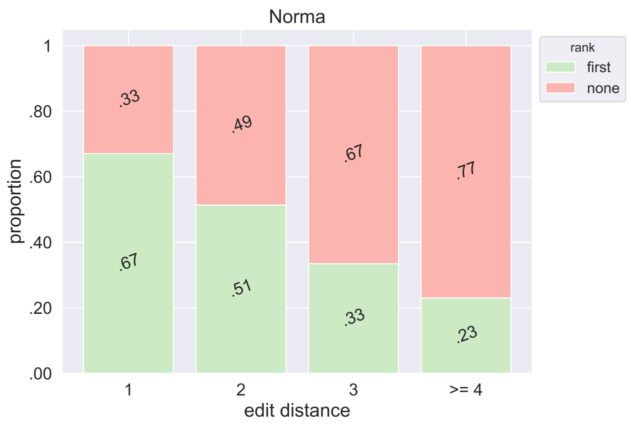
Tobias Thelen. 2010. *Automatische Analyse orthographischer Leistungen von Schreibanfängern*. Ph.D. thesis, Universität Osnabrück.

A Correction Performance per Edit Distance

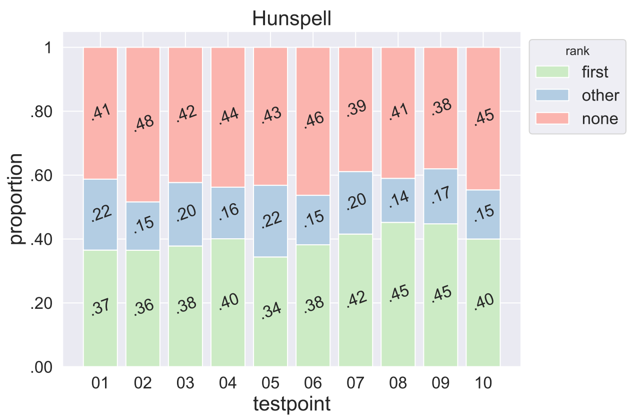


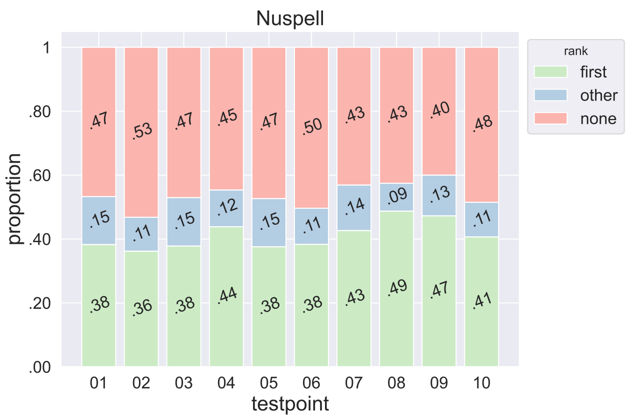


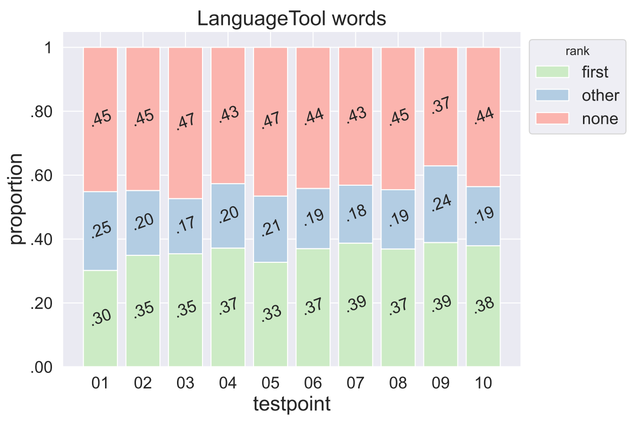


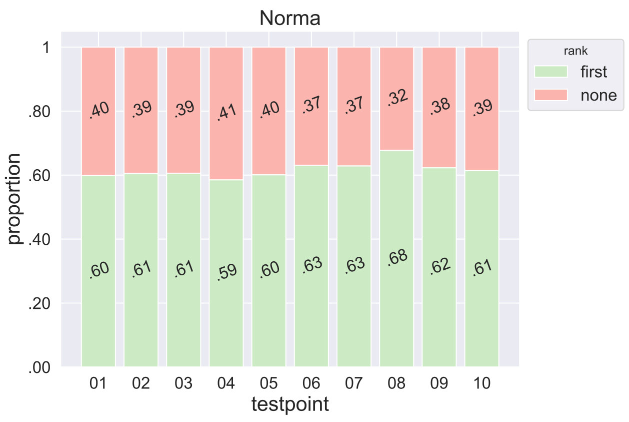


B Correction Performance per Testing Point


Hunspell
proportion
testpoint
rank
first
other
none


Nuspell
proportion
testpoint
rank
first
other
none


LanguageTool words
proportion
testpoint
rank
first
other
none


Norma
proportion
testpoint
rank
first
none