Language Teaching, Learning and Technology
6-7 Sep 2016, San Francisco, USA

Automating Multi-Level Annotations of Orthographic Properties of German Words and Children's Spelling Errors

Ronja Laarmann-Quante

Ruhr-University Bochum
laarmann-quante@linguistics.rub.de

Abstract

This paper presents the automatic annotation of orthographic properties of German words and spelling errors in texts of German primary school children according to a new multi-layered annotation scheme [1]. The scheme is closely linked to the principles of the German writing system and is supposed to allow the pursuit of new research questions concerning the relationship between spelling errors of competent and less competent spellers and the regularities of the German graphematic system. A novelty of the automatic annotation is that it takes an intended, correctly spelled word as input and applies a set of rules to generate a list of error candidates containing systematic spelling errors. As a further novelty, the annotation of additional word- and error-related properties is presented such as whether the spelling error changes the word's pronunciation and whether a spelling can be derived from a related word form. This gives rise to more detailed analyses of the errors but also allows us to develop an application for learners that generates automatic advice for the correct spelling. A first evaluation shows that the automatic annotation of the presented categories and features can come close to human annotations.

Index Terms: orthography, spelling errors, annotation, automation, German language

1. Introduction

Learning to read and write is one of the key skills to be acquired in primary school and a key to successful participation in society. In Germany, many primary school children have problems to achieve a satisfying degree of orthographic competence before entering higher school forms. In a large-scale project, we aim to get new insights into the acquisition process by analyzing what distinguishes learners with a higher spelling competence from learners with a lower spelling competence in terms of what kinds of errors they produce (rather than just the total number of errors). If orthography acquisition is not to be seen as a pure memorizing task, it requires the detection of regularities in the writing system by implicit or explicit learning. Therefore, we argue that if orthography acquisition and spelling competence is to be analyzed, a learner's writing output should be regarded in close relation to the graphematic system in order to see to what extent these regularities have been internalized.

In [1], we present a new annotation scheme for the analysis of orthographic properties of words and spelling errors in German primary school children's writings which is closely linked to the principles of the German writing system. It features multiple layers on which different properties of a word are coded which can be relevant for a detailed analysis of a given spelling error. In order to annotate a large number of texts with this scheme, it is desirable to have the annotations carried out automatically. Automated annotations also give rise to the possibility of individualized spelling diagnoses, as has been shown in [2, 3, 4]. How the automation of our annotation scheme is achieved is the topic of this paper.

The structure is as follows: In Section 2, I will present the basic graphematic foundation of our annotation scheme. Section 3 reviews related work on the topic of automatic spelling error categorization before my algorithm is described in Sections 4 to 6. I start off with the annotations of basic properties (Sec.4), then describe how orthographic phenomena in correctly spelled words are determined (Sec.5). I give an example application of this in Section 5.2 and demonstrate how the annotated properties are used to identify which kinds of errors were committed on a given misspelled word (Sec.6.1). Sections 6.2 and 6.3 describe the automatic annotation of two further error-related features, namely the pronunciation of the misspelled word in relation to the pronunciation of the intended word (*phon_orig_ok*) and the role of morpheme constancy (*morph_const*), which is roughly whether a spelling can be derived from a related word form. In Section 7, I present a first evaluation of the automated annotations and Section 8 closes this paper with a conclusion and outlook on future work.

2. Background and Motivation

German orthography is not completely shallow, i.e. there is not always a 1:1 correspondence between sounds and letters. According to the theory of Eisenberg [5], on can identify three principles in German graphemics, which I will sketch in the following.

Firstly, some words can be spelled phonographically, i.e. with regard to grapheme-phoneme correspondences (GPC). By knowing the GPC rules /b/ → <b>, /ʊ/ → <u>, /n/ → <n>, /t/ → <t> one can arrive at the correct spelling <bunt> for the word *bunt* 'colorful', which is pronounced [bʊnt].

Secondly, there are syllabic principles overriding the GPC rules. For example, the word <kommen> ['kɔmən] '(to) come' is spelled with a doubled consonant which, according to Eisenberg, is the result of the word's syllabic and prosodic structure (= a single consonant between a stressed lax vowel and an unstressed vowel). It signals that the preceding vowel has to be pronounced short. Without the doubled consonant, the form *<komen>[1] would be pronounced ['koːmən]. Another phenomenon on the syllabic level, the *syllable-separating <h>*, in contrast, does not influence the pronunciation. It graphically marks a syllable boundary between to vowels as in <Ruhe> [ʁuːə] ('quietness') which would be pronounced the same way without the <h> but its di-syllabic nature is more easily detected with the <h>. In summary, these principles serve as

[1] asterisks mark an orthographically incorrect spelling

http://dx.doi.org/10.21437/LTLT.2016-3

reading aids: they allow to derive a word's pronunciation and prosodic structure from its spelling.

Thirdly, some spellings can only be explained with regard to a related word form. An important morphological principle is that of morpheme constancy which states that the same morphemes should be spelled in the same way, regardless of the phonetic/syllabic or prosodic context. That is, the 2nd person singular present tense form of *kommen* is spelled <kommst>, including the doubled consonant as in <kommen>, although this word form does not exhibit the structural context for consonant doubling to be required.

We developed a fine-grained and multi-layered annotation scheme for (correctly-spelled and misspelled) German words that closely reflects these regularities and principles and also provides information about a word's general linguistic properties (e.g. phoneme, syllable and morpheme structure) [1]. It is supposed to allow for detailed analyses of the nature of spelling errors. In this paper, I present the automation of the main annotation levels of this scheme (implemented in Python 3).

The approach differs from existing approaches reviewed in Section 3 in that it tries to minimize probabilistic steps such as the alignment of the original and the intended spelling before anything is known about the relationship between the two.

Furthermore, this will be the first paper about the automatic examination whether morpheme constancy has an impact on the correct spelling. For instance, <Hund> 'dog' and <und> 'and' are pronounced [hʊnt] and [ʔʊnt], respectively. Unlike the word <bunt> shown above, they are not spelled phonographically, i.e. not *<Hunt> and *<unt> but with the grapheme <d>, which, according to the GPC rules, stands for the voiced plosive [d]. There is a phonological process called *final devoicing* in German which neutralizes the distinction between voiced and voiceless obstruents in the syllable coda [6]. Hence, you will never "hear" a voiced obstruent in word-final position. Nevertheless, the correct spelling <Hund> can be derived: if the learner knows how to spell the plural form <Hunde> [hʊndə], and is aware of the principle of morpheme constancy, then he or she can deduce that <Hund> also has to be spelled with a <d>. In contrast, the conjunction *und* has no related word forms, so that its spelling does not come about because of morpheme constancy but is simply called "irregular" [6] and the <d> has to be memorized. Being able to automatically detect if morpheme constancy plays a role or not can help to better evaluate a writer's errors (e.g. by asking "could the learner have arrived at the correct spelling by deriving it from a related word form?"). This also gives rise to the possibility of generating better automated spelling tips. For instance, for the misspelling *<Hunt>, advice could be to think of related word forms whereas the advice for correctly spelling <und> would have to be to simply memorize the <d>. So far it has been said that such a distinction is technically not possible[2] or that the distinction was ignored because it raises complex problems [7, p. 99].

3. Related Work

In this section, I will briefly review the work that has been done on automatic spelling error annotation of German primary school children's texts. For a broader view on learners' errors in different language contexts and their automatic processing, see for example [8].

[2]found on the website `http://ktc.dh-karlsruhe.de/descrCat/MOR_KA_AV.html` that belongs to the paper in [4]

Two researchers or groups of researchers have looked into automatic spelling error annotation for German, namely Tobias Thelen [9, 7] and Kay Berkling and colleagues [10, 4].

The approaches by Thelen [7] and Berkling et al. [10] both have in common that they first align original and target spelling (that is, the spelling produced by the learner and the intended, correct spelling) and if an original character(-sequence) differs from its corresponding target character(-sequence), they determine which error category is present. One major difference between the approaches (besides the different error categories they look at) is how the alignment is carried out. Thelen aligns on the character level, using Levenshtein distance which takes into account information about phonetic features that were inferred from the characters. Berkling et al. use the speech synthesis system MARY [11] to obtain the phonemes for both original and target spelling and align those using phoneme features. Based on the phoneme alignment, characters are split into graphemes and aligned.

Aligning original characters and target characters this way can be problematic. Consider the misspelling *<sten> for <stehen>. A simple Levenshtein-based alignment on the character level yields two equally likely outcomes:

(1) a.

orig	s	t			e	n
target	s	t	e	h	e	n
pronunciation	ʃ	t	eː		ə	n

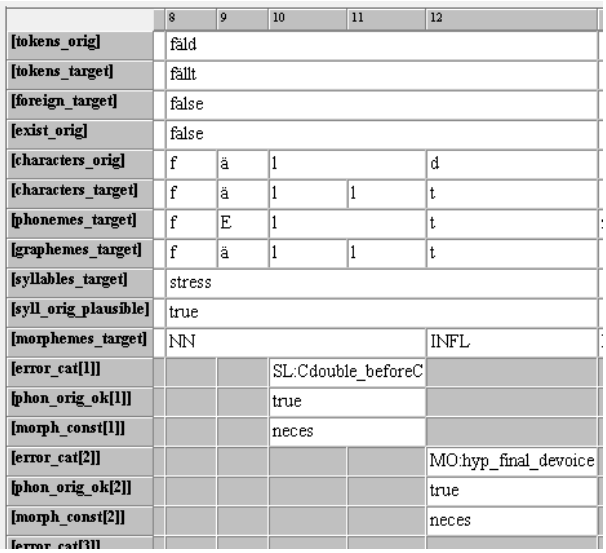
b.

orig	s	t	e			n
target	s	t	e	h	e	n
pronunciation	ʃ	t	eː		ə	n

Speakers of German know that <stehen> is pronounced [ʃteːən] only in formal contexts and that the common pronunciation is [ʃteːn]. More generally, we know that a schwa is often dropped in pronunciation so that alignment (1b) is more plausible where the character <e> in the original spelling represents the full vowel [eː] rather than the schwa. Using the phoneme-alignment method from [10] for this example would be misleading, though. MARY returns [ʃtən] as the pronunciation of *<sten> because it depends on rules rather than a lexicon that is used for known words. This pronunciation would suggest alignment (1a) and lead to a different error category (if omission of full vowel and schwa are distinguished which they are in [10]).

Therefore, I propose an approach which first determines possible systematic errors on a target word which then "emit" an error candidate. Comparing the error candidates with the observed misspelling then yields the error categories and alignments automatically. This is assumed to be more robust than a-priori alignments of original and target characters. I will explain this more thoroughly in the following sections.

A similar approach has been proposed in an earlier work of Thelen [9], which he calls a "top-down" approach. He emphasizes the advantage that the correct spelling already "knows" which features are present that can be misspelled (p. 81). His concrete implementation, however, is neither completely robust nor comprehensive in terms of covered error categories. He takes the pronunciation of the target word as the basis, which was obtained from the target word string via a grammar. The outcome is a number of possible pronunciations from which the correct one has to be chosen manually. The phonemes then "emit" a number of possible graphematic realizations, one of them being the orthographically correct variant and the other ones possible misspellings, which are automatically associated



	8	9	10	11	12
[tokens_orig]	fäld				
[tokens_target]	fällt				
[foreign_target]	false				
[exist_orig]	false				
[characters_orig]	f	ä	1		d
[characters_target]	f	ä	1	1	t
[phonemes_target]	f	E	1		t
[graphemes_target]	f	ä	1	1	t
[syllables_target]	stress				
[syll_orig_plausible]	true				
[morphemes_target]	NN				INFL
[error_cat[1]]			SL:Cdouble_beforeC		
[phon_orig_ok[1]]			true		
[morph_const[1]]			neces		
[error_cat[2]]					MO:hyp_final_devoice
[phon_orig_ok[2]]					true
[morph_const[2]]					neces
[error_cat[3]]					

Figure 1: Screenshot of EXMARaLDA with our annotations

with error categories. The biggest disadvantage of this approach is that the orthographically correct variant first has to be (re-)produced from the phoneme string as well, which might be especially problematic for non-native words. If this failed, further analyses would not be possible[3]. I want to overcome these deficits by taking the correctly spelled word as input and only use phoneme information as additional clues where necessary. This has the further advantage that also error candidates can be produced which do not have anything to do with the pronunciation (e.g. the confusion of similar looking letters such as <b> and <d>).

4. Basic Property Annotations of Target Words

Each annotation according to our scheme consists of several layers. The annotations are stored in an XML-format called *LearnerXML* that we describe in [1]. They can be visualized in the *Partitur-Editor* of the tool EXMARaLDA[4] [12, 13]. Figure 1 exemplarily shows the full annotation of the misspelling *<fäld> for <fällt> 'he falls'.

For the automatic annotation described in this paper, the original word produced by the learner and the target word, which is the word form that the learner most probably had in mind, are given as input. Both words are split up into single characters to allow for a precise localization of the error in a word. In this section, I will describe how we automatically obtain the annotations for the layers **phonemes**, **graphemes**, **syllables** and **morphemes**, which relate to the target word. Section 5 then describes how these layers are used to determine orthographic properties (defined as *possible* error categories) of the target words and Section 6 explains how the categories of the actually *committed* errors are determined given an original word and the target word and how original and target characters are aligned. The automatic determination of the impact of a spelling error on the word's pronunciation (layer **phon_orig_ok**) and the role of morpheme constancy in a spelling (layer **morph_const**) are described in Sections 6.2 and 6.3, respectively. Not yet implemented are the layers **foreign_target** (is the target word a foreign word?), **exist_orig** (does the misspelling result in another existing word form?) and **syll_orig_plausible** (does the syllable in the original spelling not violate any graphotactic constraints?).

For any given word, its phonemes, syllables and morphemes are obtained from the web service *G2P* of the Bavarian Archive of Speech Signals (BAS)[5] [14, 15]. An example output for the word <fröhlich> 'happy' looks like this [6]:

(2)

input string:	`fröhlich`
phonemes w/ syll.:	`f r ' 2: . l I C`
POS:	`ADJD`
morphemes:	`fröh lich`
morpheme classes:	`ADJ SFX`

The information obtained are processed and converted to annotations in the following way:

Phonemes and phoneme-corresponding units (PCUs) Firstly, phonemes (without syllable boundaries and stress marks) are aligned to characters. The most robust way turned out to be the use of a statistical approach to string alignment via weighted Levenshtein distance[7] plus additional rules instead of the alignment function of the *G2P* web service, which often had problems aligning words with <x>/[ks] or <z>/[ts]. The alignment script was trained on 36,790 words from childLex, the German Children's Book Corpus [17][8]. The output is a 1:1 (or 1:0 or 0:1) alignment of phonemes and characters.

Secondly, with the help of rules it is determined which sequence of characters corresponds to a phoneme. I call these sequences *phoneme-corresponding units (PCUs)*[9]. PCUs which are not trivial 1:1 alignments between characters and phonemes are multi-graphs (e.g. <ch> ↔ [ç], [x] or [k], <sch> ↔ [ʃ]), diphthongs (<ei> ↔ [aɪ]), long vowels (<ah>, <ee>, <ie> ↔ [aː], [eː], [iː] etc.), doubled consonants (<pp>, <tt>, <ck> ↔ [p], [t], [k] etc.), vocalized *r* in <er> ↔ [ɐ] as well as <z> ↔ [ts], <x> ↔ [ks], <qu> ↔ [kv] and empty string ↔ [ʔ]. The PCUs and phonemes are taken over into LearnerXML.

[3] Thelen claims that for all 500 test words the orthographically correct variant was produced but the list (pp. 154ff) shows that for some it was not (e.g. <genug> 'enough' was not produced, only *<genuch>). Either the displayed list is incorrect or there were indeed unmentioned exceptions.

[4] `www.exmaralda.org`.

[5] `https://webapp.phonetik.uni-muenchen.de/BASWebServices/#/services/Grapheme2Phoneme`

[6] Phonemes are given in SAMPA notation as specified under `http://www.phon.ucl.ac.uk/home/sampa/german.htm`, POS tags follow the STTS tagset [16] and morpheme classes are specified under `http://www.phonetik.uni-muenchen.de/Bas/BasWebservicesG2P.html`.

[7] script created by Marcel Bollmann; `https://github.com/mbollmann/`

[8] `https://www.mpib-berlin.mpg.de/de/forschung/max-planck-forschungsgruppen/mpfg-read/projekte/childlex`; The words chosen were all listed word forms of all lemmata that occurred at least 25 times in the corpus, ignoring letter case. For the training, pairs of words (=sequences of characters) and their pronunciation (=phoneme strings) were presented to the algorithm without prior alignment of those.

[9] In some approaches, graphemes are defined as dependent on phonemes and in these definitions, *PCUs* and *graphemes* would be the same [18]. However, I follow Eisenberg's grapheme definition [5] so that these entities are different.

Example (3) illustrates the alignment steps (<_> stands for the empty string, phonemes are given in SAMPA):

(3)
```
|f|r|ö|h|l|i|c|h|  →  |f|r|öh|l|i|ch|
|f|r|2|:|l|I|_|C|      |f|r|2:|l|I| C|
```

Graphemes Graphemes are defined following the definition in [5] in that the only multi-character graphemes are <ch>, <sch>, <ie> and <qu>. They are determined with the help of PCUs. If the character sequence <sch> corresponds to a PCU as in <F|l|a|**sch**|e> [f|l|a|**ʃ**|ə] 'bottle' it is a grapheme (vertical bars indicate PCU boundaries). If it spans over several PCUs as in <b|i |s**s**|**ch**|e|n> [b|ɪ|**s**|**ç**|ə|n] 'a little', it is not.

Syllables After the determination of PCUs, syllable boundaries and stress marks are re-inserted into the phoneme string and with help of the alignment between characters and PCUs, syllable boundaries can be indicated on the character level. The type of the syllable is determined as follows: each syllable that contains a stress mark is classified as *stressed*, each syllable that has [ə] or [ɐ] as its nucleus is a *reduced* syllable and each other one an *unstressed* syllable.

Morphemes Morpheme boundaries and classes on the character level are given directly in the BAS output and are simply taken over as annotations.

5. Annotating Properties of Correctly-Spelled Words

Our annotation scheme was designed in a way that the error categories can be seen as the orthographic properties a target word possesses[10]. For instance, *SL:Cdouble_interV* indicates that the word contains a doubled consonant in intervocalic position (*SL* tells that this is a syllabic phenomenon). If a learner omitted this, *SL:Cdouble_interV* is the respective error category for this. Other annotation schemes do not strictly categorize by orthographic phenomena. [19], for instance, which is the annotation scheme used in [10] has a category for vocalized <r>, which possibly applies to <dort> [doɐt] 'there', but vocalized <r>s in a reduced syllable, in a free or bound morpheme are not coded as such but assigned to categories like *free grammatical morpheme*. In our scheme, the syllables and morphemes are coded on different layers so that we have all pieces of information (phenomenon + syllable and morpheme type) available at the same time. This allows to make statements about which words contain a vocalized <r> but also to distinguish between the syllabic or morphological contexts if this turns out to be sensible. In the following, I will explain how the orthographic properties are determined automatically.

5.1. Procedure

The approach to annotating orthographic properties is rule-based. As explained above, properties are equal to possible error categories which are annotated at the character level (usually spanning over a PCU). For each category in our annotation scheme, detailed rules exist which state which phonetic/syllabic/morphological context is necessary for a character to be annotated with this category[11]. Here is the example for the category *voc_r* which marks a (potentially) vocalized <r>:

(4) Category **voc_r** applies to:

a. every PCU corresponding to [ɐ]
e.g. <L|eh|r|**er**> [l|eː|ʁ|**ɐ**], <d|o|**r**|t> [d|ɔ|**ɐ**|t]

b. every PCU corresponding to [ʁ] which is in the syllable coda (according to the German pronunciation rules [20, 21] vocalization is possible or even mandatory in this position); it may have technical reasons that some <r> in this position were not transcribed as [ɐ] by the BAS web service.

c. every PCU corresponding to [ʁ] in a reduced syllable that ends with <en> This is supposed to capture cases like the colloquial pronunciation of <fahren> which is [faːɐn] insead of [faːʁən]. If the schwa is omitted, the <r> moves to the syllable coda and is vocalized.

5.2. Application Example

Knowing the orthographic phenomena occurring in a word has several applications. First of all, in order to assess a learner's spelling competence it is not enough to look at the committed errors but also to set this in relation to the errors he/she could have committed but did not (the so called *base rate* [19]). Secondly, one can determine which phenomena learners are faced with e.g. in school books and see how this relates to the development of their spelling competence [22]. To detect specific orthographic phenomena can also be of relevance for psycholinguists who want to create stimuli for orthographic experiments.

In this section, I want to briefly introduce how the annotations with our scheme can be used to automatically detect words which can be spelled phonographically, e.g. only following grapheme-phoneme correspondence rules. This strategy, i.e. "to spell a word as it is pronounced" is at the core of a currently widely-used teaching method for learning to write (and read) in German primary schools, which is called *Lesen durch Schreiben* ('reading through writing') [23]. In the center of this method is a so-called *onset table* ('Anlauttabelle'). In such a table, graphemes are depicted next to an object whose name starts with the phoneme that this grapheme corresponds to (e.g. <sch> → *Schere* 'scissors'). The idea is that learners identify each sound in a word and search for the grapheme representing this sound in the table, thereby consecutively constructing the word's spelling. It would be interesting to see which proportion of words in German can actually be correctly spelled this way. To find this out, I determined the annotation categories which indicate that the word in question contains a phenomenon that is not fully GPC-compliant given standard German pronunciation and the GPC rules in [5]. These categories are:

PGI:literal
: spellings of phoneme combinations that differ from the phonographic spelling of their constituent phonemes (e.g. [ʃp] is spelled <sp> and not *<schp> as in <**sp**ielen> '(to) play'

[10] There are also categories referring to hypercorrections of a phenomenon, which would in contrast indicate the absence of a particular phenomenon in the target word and some categories do not refer to orthographic categories as such but can only sensibly be applied as error categories, e.g. confusing similar looking letters such as <d> and <b>.

[11] Not yet implemented are only the categories *PGI:de_foreign*, *MO:morph_between* and *PGII:diffuse*.

PGI:repl_unmarked_marked
spellings with letters (or combinations) that do not appear in the basic GPC rules (e.g. <v>, <äu>)

SL:Cdouble_interV, SL:Cdouble_beforeC, SL:Cdouble_final
consonant doubling, e.g. <Fa**ll**e> 'trap'

SL:separating_h
syllable-separating <h>, e.g. <se**h**en>, '(to) see'

SL:Vlong_ie_ih, SL:Vlong_ie_i, SL:Vlong_ie_ieh, SL:Vlong_single_double, SL:Vlong_single_h
vowel-lengthening <h>, vowel doubling and <i> pronounced long, e.g. <T**i**ger> 'tiger', <F**ah**ne> 'flag', <B**oo**t> 'boat'

SL:ins_schwa
phonological schwa-elision before the syllabic consonants /l/, /m/ and /n/, e.g. <les**e**n> [leːzn̩] '(to) read'

SL:voc_r r-vocalization, e.g. <Lehr**er**> [leːʁɐ] 'teacher'

MO:final_devoice final devoicing, e.g. <Wal**d**> [val**t**] 'forest'

MO:final_ch_g
g-spirantization, e.g. <traur**i**g> [tʁaʊʁɪ**ç**] 'sad'

MO:morph_in
morpheme boundaries, e.g. <Han**d**+**t**uch> [han**t**uːx] 'towel'

I applied the automatic annotation to the 36,790 frequent word forms from childLex that were also used for training the character-phoneme alignment script. All tokens that contained at least one of the categories above were marked as non-phonographic spellings[12]. The results were the following: only 3,149 tokens (8.6%) did not contain any of the above categories, i.e. could be spelled purely phonographically. A manual inspection of a subset of 200 tokens revealed 10 disagreements between the manual judgments and the automatic output. Only 4 of them can be attributed to errors on the automatic tool's side (mainly wrongly assuming final devoicing with the character sequence <ng>, which stands for [ŋ]), the other disagreements came about because the human annotation overlooked a non-phonographic spelling (this task is not easy to carry out for literate persons because orthographic knowledge has to be blocked out). Judging from this small experiment, automatically detecting phonographic spellings with our annotations seems to be possible and quite reliable. It is also easy to adjust the criteria for phonographic spellings. The categories *PGI:literal* and *SL:ins_schwa* could be argued to be too strict in that the spelling of certain phoneme combinations could be easily assigned to own GPC rules and the schwa does not have to be omitted in pronunciation. With these categories removed, 23.5% phonographic spellings in 36,790 words were observed, which is still clearly the minority and questions the central role of phonographic spellings in the mentioned didactic approach.

6. Annotating Spelling Errors

6.1. Error Categories

If you know both the original spelling and the target spelling, and you already know which error categories can possibly apply on the target word, it is straightforward to find out which errors were in fact committed. To this end, not only the possible error categories are determined but for each category one (or more) error candidates are constructed, which is what the word would look like if this error had in fact occurred. Take the word <fällt> '(he) falls' from Figure 1. (5) shows possible error categories and the corresponding error candidate. The vertical lines indicate the PCU boundaries:

(5)

PG:repl_marked_unmarked	**v**|ä|ll|t, **ph**|ä|ll|t
SL:Cdouble_beforeC	f|ä|**l**|t
SL:rem_Cdouble_afterC	f|ä|ll|**tt**
MO:hyp_final_devoice	f|ä|ll|**d**

If the original spelling corresponds to one of the error candidates, the error category can be seen directly. Of course it is possible that more than one error occurs on a word. If the original spelling does not correspond to one of the simple error candidates, combinations of those have to be created. To achieve this, the PCUs that are affected by the possible errors are extracted (the ones printed in bold in the example above). Then, all possible combinations of correct and erroneous PCUs in a word are computed. (6) shows this for the word <fällt>. Each column represents a PCU and lists all candidates that are obtained from possible errors:

(6)

PCU1	**PCU2**	**PCU3**	**PCU4**
f	ä	ll	t
v	e	l	dt
ph	äh		th
ff	ää		tt
w	a		d
vv[13]	*eh*, *ee*		*dd*
ww	*ah*, *aa*		

Traversing from left to right, taking one candidate from each column, one can construct all possible error candidate words, e.g. <fältt>, <velld>, <fald> or <fäld>. The latter one is the original spelling we observed in Figure 1. We know that <f> and <ä> are correct, and from (6) we know that <l> stems from the error category *SL:Cdouble_beforeC* and <d> from the error category *MO:hyp_final_devoice*. We also know how the original and target word have to be aligned because the target word has "emitted" the original word. This allows a precise alignment where a character-based Levenshtein algorithm would fail as shown for the misspelling *<fältt> in (7):

(7) a. Levenshtein algorithm

orig	f	ä	l	t	t
target	f	ä	l	l	t

b. Our algorithm

orig	f	ä	l		t	t
target	f	ä	l	l	t	

We can now also explain more concretely how we arrive at the desired alignment for the example from Section 3, the misspelling *<sten> for <stehen>: Both the <h> and the second <e>, which represents the schwa sound, emit as error candidates that they can be omitted (categories *SL:separating_h* and *SL:ins_schwa*), the first <e>, which represents the sound [eː] has no omission as error candidate. Hence, it is clear that the <e> that was omitted in *<sten> was more plausibly the one representing the schwa and not the [eː].

[12] Furthermore, the feature *phon_orig_ok* discussed in section 6.2 had to be *true* to rule out all phenomena which only occur with a colloquial pronunciation as in example (4c).

[13] candidates in italics are obtained by combining two types of errors within one PCU

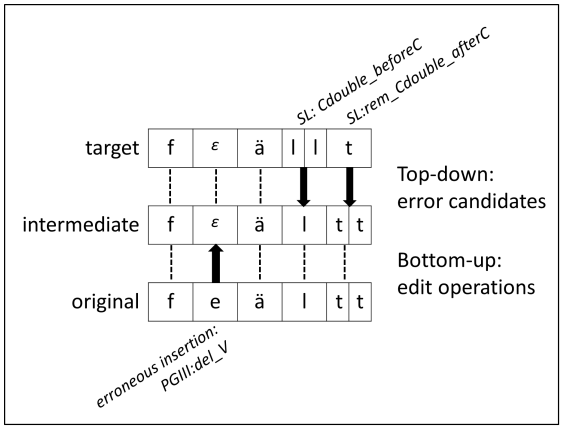


Figure 2: Illustration of the top-down-bottom-up procedure for annotating spelling errors (ε denotes the empty string)

Eventually, there are some kinds of errors which cannot be predicted and which I call *unsystematic errors*, for instance a seemingly random insertion *<f**e**ällt> or replacement *<f**u**llt>. Unsystematic errors are all errors for which our annotation scheme has no systematic category. As in [9], they are only described with the basic edit operations *insertion*, *deletion* and *replacement* plus *permutation*. To capture them and align them correctly, a similar approach as by Thelen [9] is taken, which he describes as "top-down-bottom-up". Assume the original spelling for <fällt> is *<feältt>. First of all, we try to capture all the systematic errors in the word. This is done by constructing error candidates as described above (top-down) but none of these candidates corresponds exactly to the original word. Therefore, the error candidate with the smallest Levenshtein distance to the original word is chosen, which is in this case <fältt>, and all further deviations between the two are treated as unsystematic errors. Here only the erroneously inserted <e> is left as an unsystematic error, which is the desired outcome. To capture the inserted <e> correctly, the 'classical' bottom-up approach is taken in that the intermediate error candidate <fältt> and the original spelling <feältt> are aligned using the Levenshtein algorithm[14] and the edit operations including permutation are determined based on this alignment. The final alignment from the original spelling to the target spelling is carried out in a transitive way: the original spelling is aligned with the intermediate spelling and this is in turn aligned with the target spelling, so the alignment can be passed on. Figure 2 illustrates the procedure.

6.2. Relation between Spelling Errors and Pronunciation (phon_orig_ok)

In our annotation scheme, we do not only code basic properties of words and error categories, but also further features pertaining to the errors. One of them is *phon_orig_ok* which codes to what extent the spelling error influences the pronunciation of the word. In our scheme, there are three possible values: *true* means that even with the spelling error the word is still possibly pronounced in the same way as the target word (e.g. *<faren> for <fahren>, both [faːʀən]). *false* means that the pronunciations differ (e.g. *<komen> for <kommen>: ['koːmən] vs. [kɔmən]) and *coll* that the resulting pronunciation is a colloquial or regional variant (e.g. *<Tellefon> for <Telefon>: [tɛːləfoːn] vs. [teːləfoːn]). With the help of this feature, we want to examine systematically how learners treat the relation between the spelling of a word and its pronunciation. One hypothesis would be that understanding and obeying this relation is a requirement for successful spelling. Following from this, learners with overall higher spelling skills (e.g making fewer errors altogether) are aware of this relation and make relatively fewer errors that change the pronunciation of a word than overall poorer spellers. A similar feature has been applied in the spelling error annotation by [24] for English data. For German, information on pronunciation could only be found in Thelen's scheme [7] so far with a feature named "phonetically plausible". Phonetic plausibility is only coded very broadly in this approach, though. For instance, two consonants are judged as similar if they are both nasals or plosives or fricatives. The <g> for <k> in *<gomen> for <kommen> is marked as similar there, while our coding would make clear that the pronunciations differ.

With the approach in [10], where the phonemes of the original and target spelling are aligned prior to the error analysis, one could simply check whether they are identical to arrive at the value for our feature *phon_orig_ok*. This works well for example for the pair *<fäld>/<fällt> because the BAS web service *G2P* and MARY return the same phonemes [fɛlt] for both spellings. However, as was already demonstrated in Section 3, the grapheme-phoneme conversion for non-existing German words can be flawed so that the spellings appear to have different pronunciations although in fact they do not.

Our algorithm, in contrast, concentrates on the error categories: the value of *phon_orig_ok* is determined when the possible errors and error candidates for each word are constructed. The rules which state when an error category applies and what the error candidate looks like also contains the information whether the pronunciation of the error candidate PCU will be different from the target PCU. In most cases, the value for *phon_orig_ok* is the same for all instances of a category (e.g. *MO:final_devoice* (final devoicing) is always *true*; insertion of <h> after a short vowel (*SL:rem_Vlong_short*) is always *false*). For a few categories, further cases are distinguished, e.g. for the category *SL:Cdouble_interV*: the omission of the doubled consonant usually leads to a change in pronunciation (see above), hence *phon_orig_ok = false*. However, there are some words with a doubled consonant which do not have a trochaic stress pattern that triggers the German *Schärfungsschreibung* as explained in Section 2. Therefore, there is a subrule which states that if not the first but the second syllable is stressed, the omission of consonant doubling does not change the pronunciation, hence *phon_orig_ok = true* (e.g. *<alein>/<allein> [ʔaˈlaɪn] 'alone').

6.3. Role of Morpheme Constancy (morph_const)

For each error, we code the role of morpheme constancy for arriving at the correct spelling. Specifically, the guiding question is "Do I have to know how a related word form is spelled in order to arrive at the correct spelling?" given that the pronunciation of the erroneous spelling and the target spelling are phonetically equivalent. The feature *morph_const* can take one of four values: *neces* means that morpheme constancy is a necessary reference, for instance to arrive at the doubled consonant in <kommst> (inherited from <kommen> as explained above). We also defined that the spelling of bound morphemes falls under morpheme constancy. For instance, in the exam-

[14] python-Levenshtein obtained from `https://pypi.python.org/pypi/python-Levenshtein/`

ple *<fäld> for <fällt> in Figure 1, the <d> for <t> is a hypercorrection of final devoicing (possibly constructed in analogy to e.g. <Feld> 'field'). We say that *morph_const = neces* here because if the learner had recognized that this character is the marking for 3rd person singular present tense, which is always <t> and not <d>, he or she could have deduced the correct form from analogous forms like <sag-t> '(he) says, <lach-t> '(he) laughs', etc. The value *redun* ('redundant') means that a particular phenomenon could be both prosodically determined and motivated by morpheme constancy (e.g. <kommend> 'coming'). In some exceptional cases, morpheme constancy is violated, as in <Bus> 'bus', which does not inherit the doubled <s> from the plural form <Busse>. If the learner wrote *<Buss> instead, the value of the feature *morph_const* is *hyp* ('hypercorrection'). Finally, if morpheme constancy does not play a role or is not applicable because there are no related word forms (as in the case of <und> explained above), the value is *na*.

These values are determined automatically in the following way: Firstly, in several cases, the error category already rules out that morpheme constancy plays a role. For instance, all errors of the category *PGI:literal* (spelling of a particular fixed phoneme combination was not obeyed, e.g. *<schpielen> for <spielen> '(to) play') have *morph_const = na*. On the other hand, all instances of the category *MO:morph_in* (omission of graphemes at morpheme boundaries, e.g. *<Hantuch> for <Hand+tuch> 'towel') are tagged as *morph_const = neces* because it always involves the identification of morphemes.

In some error categories, different values of *morph_const* are possible. As explained in Section 2, both *<Hunt> for <Hund> and *<unt> for <und> are categorized as *MO:final_devoice*. However, the <d> in <Hund> can be said to be inherited from the form <Hunde> (*morph_const = neces*), while the <d> in <und> cannot be motivated (*morph_const = na*). To get such a distinction, the automatic annotation proceeds as follows: First of all, the morpheme classes obtained from the BAS web service are divided into inflecting (e.g. nouns and verbs) and non-inflecting word classes (e.g. pronouns and conjunctions) and affixes (bound morphemes). In a category where morpheme constancy might play a role, it is checked if the grapheme (or PCU) in question is part of a bound morpheme (as in the *<fäld> example above) or if it is in the final position of an inflecting morpheme[15]. In both cases *morph_const* is set to *neces*. To appear in an inflecting morpheme is supposed to model the possibility of a related word form from which the correct spelling can be derived. It is not checked whether such a word form does in fact exist, though. That the grapheme in question has to be in morpheme-final position is necessary for cases such as <O**b**st> [o:**p**st] 'fruit', where the <b> occurs in an inflecting word class (noun) but where no related word form (e.g. genitive <Obstes>) would make a [b] perceptible.

Another heuristic is used to distinguish between spellings with <ä> and <äu> that are morphologically determined (e.g. <Äpfel/Apfel> 'apples/apple', <Mäuse>/<Maus> 'mice/mouse') and those which are not (at least not synchronically, e.g. <Mädchen> 'girl', <Säule> 'pillar'). If you replace the <ä> with an <a> and remove any final <e/er/en/ern> from the string (modeling suffixes) and the resulting word form exists in a given lexicon, then the word form is assumed to be morphologically determined (*morph_const = true*), otherwise it is classified as *morph_const = na*.

[15]The error category determines what exactly is checked: Sometimes only bound morphemes are relevant and not the appearance in an inflecting word form.

7. Evaluation

In this section, I will present first evaluation results of the automatically obtained annotations for a) the error categories, b) the level *phon_orig_ok* and c) the level *morph_const*. The annotations were carried out on 11 texts (866 target tokens) from the corpus described in [25], where children from grades 2-4 were asked to write down the story shown in a sequence of six pictures. The same texts were manually annotated by three humans for comparison (they were a subset of the annotation experiment reported in [1]).

Table 1 lists the agreement figures[16]. The figures for the error categories are based on the number of annotations and the figures for the features *phon_orig_ok* and *morph_const* only on those annotations where the error categories were the same for all three annotators and the system. Since the system determines the value of *phon_orig_ok* and *morph_const* based on the error category, a comparison with human results is only sensible where this matches. In the lower part of the table, figures are given for the case that annotations with the error category *SN* were not taken into account. This comprises phenomena beyond single word spelling such as capitalization and writing words together or separately, which are trivial to detect without the procedure presented in this paper and where *phon_orig_ok* and *morph_const* are always *true* and *na*, respectively.

The average agreement for error categories between the system and the annotators is not much lower than the agreement among the human annotators themselves, which indicates a good performance of the automatic algorithm. For *phon_orig_ok* and *morph_const* one can see bigger differences between the system and the individual annotators with annotator 1 appearing as an outlier. It shows that the annotation task does not yield uniform results among human annotators either and that stricter and more transparent guidelines are required. Once we have enough reliably hand-annotated data, we want to perform another evaluation of the system's performance based on gold data so that a "correct/incorrect" decision can be made[17]. I will now analyze the major sources of disagreement which are attributable to the algorithm or implementation:

error categories One bigger problem was that I had to put a constraint on the number of generated error candidates. If many single errors were possible on a given target word, the handling of all possible combinations would be computationally too complex. Therefore, I decided that combinations would only be computed if the target word featured less than 20 single possible errors. In this case, all but one systematic errors would be (falsely) treated as unsystematic errors (e.g. simply *insertion* or *replacement*). For a new version of the system, I changed this behavior: If more than 20 single errors are possible, all combinations are computed but without the category *SL:Cdouble_afterC*, which captures doubled consonants in word-initial position or after other consonants. This category produces a huge number of error candidates but is needed only rarely. So far, leaving this category out seems to suffice to keep

[16]thanks to Stefanie Dipper for the computation of the agreement with the software R and the package "irr", `https://cran.r-project.org/web/packages/irr/`

[17]However, there will always remain some cases where more than one interpretation is possible e.g. in the case of *<gächt> for <gebracht> 'brought', where the <ä> could both represent the <e> or <a> in the target spelling.

	Level	Size	Sys - Anno1		Sys - Anno2		Sys - Anno3		Average		Human	
			perc.	κ	perc.	κ	perc.	κ	perc.	κ	perc.	κ
complete	error cat.	261	73.18%	.71	80.84%	.79	81.99%	.80	78.67%	.77	77.39%	.82
	phon_orig_ok	227	75.45%	.53	84.00%	.69	81.60%	.66	80.35%	.63	78.03%	.72
	morph_const	227	70.00%	.38	78.23%	.51	90.40%	.80	79.54%	.57	70.99%	.55
w/o SN	error cat.	170	65.88%	.64	75.29%	.74	75.29%	.74	72.16%	.71	72.94%	.79
	phon_orig_ok	145	75.45%	.53	83.87%	.69	81.45%	.65	80.26%	.63	78.03%	.72
	morph_const	145	70.00%	.38	78.05%	.51	90.32%	.80	79.46%	.57	70.99%	.55

Table 1: Inter-annotator agreement (raw percent and Fleiss' κ) for error categories and the features *phon_orig_ok* and *morph_const* for all annotations and for annotations excluding the category *SN*. *phon_orig_ok* and *morph_const* are based only on annotations where the same error category was annotated by all three annotators (and the system, respectively). Figures are given for the pairwise agreement between the system and one annotator, the average, and the agreement among the human annotators only.

the computational complexity in an adequate range and yield better annotation results. Furthermore, some errors in the implementation could be found, which are corrected in a new version such as the identification of the permutation of graphemes.

phon_orig_ok A larger proportion of the disagreements is due to a systematically different behavior for the category *SL:Cdouble_final* (e.g. *<dan> for <dann>). The human annotators consequently tagged this as *phon_orig_ok = false* because of a change in vowel length from short to long (<dann> = [dan], *<dan> = [da:n]) but the system tagged it as *phon_orig_ok = true*. According to Eisenberg [26, pp. 96f], the vowel in a monosyllabic word with a single grapheme in the coda has to be read as long (e.g. <Schal> 'scarf', <Rad> 'wheel'). There are exceptions, however, namely a number of words from closed word classes and words with a multi-letter grapheme in the coda where the vowel is read short (e.g. <in> 'in', <zum> 'to the', <man> 'one', <Busch> 'bush'). Hence, the automatic system is not completely wrong in predicting that a monosyllabic word ending with a single consonant could be pronounced with a short vowel as well. This behavior could easily be changed in future versions, though.

A further difference between the human and automatic annotation is that humans apply morphological knowledge when judging the pronunciation of a word while the current automatic algorithm does not. This can be seen in a misspelling like *<knalte> for <knallte> 'banged'. The automatic system judges that the omission of the doubled consonant before other consonants does not lead to a change in pronunciation (in particular: no change in the length of the preceding vowel from short to long). In fact *<knalte> does not differ in syllable structure from the word <kalte> 'cold' which is pronounced [kaltə], i.e. with a short vowel. However, the human annotators intuitively considered that the <-te> in *<knalte> is an inflecting morpheme and that the resulting stem <knal> would be pronounced with a long vowel (which is not changed by affixation).

A few other cases could not be captured by the automatic algorithm because they were "exceptions" from a general rule. For instance, the spelling *<Tellefon> for <Telefon> 'telephone' results in a common colloquial pronunciation of the word, which the current algorithm that focuses on the kind of error that was committed, could not predict. To capture such colloquial cases, one could think about applying machine learning approaches once we have enough gold data to extract the relevant contexts from.

morph_const Some errors in the implementation became apparent (e.g. a morpheme class was mistakenly not assigned to be non-inflecting or a necessary condition was not implemented for a category), which can easily be remedied. Secondly, as was expected, the heuristics failed sometimes, for instance it was not automatically detected that the <ä> in <Hälfte> 'half (noun)' can be explained morphologically (with reference to <halb> 'half (adj.)').

8. Conclusion

In this paper, I presented the automatic annotation of German texts according to a new multi-layered scheme which can be used for the annotation of orthographic properties of words and committed spelling errors. The proposed approach for error annotation is supposed to be very robust in that spelling errors are interpreted on the basis of possible errors that can occur in a word. Furthermore, it allows to automatically analyze the orthographic properties of German words in general as was shown exemplarily for obtaining words that can be spelled phonographically. I also presented the first automation of the question whether a spelling error changes a word's pronunciation and whether the spelling can be derived from a related word form. A first comparison with human annotations showed promising results which could be improved with some adjustments in the system.

Taken together, the annotations provide a potentially powerful frame for the analysis of spelling errors on various levels. In future work, we want to arrive at a large corpus of texts produced by primary school children with detailed error annotations and use this to carry out statistical analyses about the relationship between spelling errors, spelling competence and regularities of the German writing system. On the other hand, we would like to use the automated error annotation to develop an application which can assist learners in orthography acquisition in that it does not only provide an automated error diagnosis on a freely written text but can also generate advice how to arrive at the correct spelling. A collaboration with educational research and didactics in this endeavor would be very fruitful.

9. Acknowledgements

This research is part of the project *Literacy as the key to social participation: Psycholinguistic perspectives on orthography instruction and literacy acquisition* funded by the Volkswagen Foundation as part of the research initiative "Key Issues for Research and Society".

10. References

[1] R. Laarmann-Quante, L. Knichel, S. Dipper, and C. Betken, "Annotating spelling errors in German texts produced by primary school children," to appear in Proceedings of the 10th Linguistic Annotation Workshop (LAW), Berlin, Germany, 2016.

[2] K. Berkling, "A case study using data exploration of spelling errors towards designing automated interactive diagnostics," in *WOCCI*, 2012, pp. 97–103.

[3] ——, "A non-experts user interface for obtaining automatic diagnostic spelling evaluations for learners of the German writing system," in *INTERSPEECH*, 2013, pp. 1879–1881.

[4] K. Berkling and R. Lavalley, "WISE: A web-interface for spelling error recognition for German: A description of the underlying algorithm," in *Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology*, Duisburg/Essen, Germany, 2015, pp. 87–96.

[5] P. Eisenberg, *Grundriss der deutschen Grammatik Band 1: Das Wort*, 3rd ed. Stuttgart: J.B. Metzler, 2006.

[6] T. A. Hall, *Phonologie: Eine Einführung*, 2nd ed. Berlin: de Gruyter, 2011.

[7] T. Thelen, "Automatische Analyse orthographischer Leistungen von Schreibanfängern," Ph.D. dissertation, Universität Osnabrück, 2010.

[8] C. Leacock, M. Chodorow, M. Gamon, and J. Tetreault, *Automated Grammatical Error Detection for Language Learners*, 2nd ed. San Rafael, CA: Morgan & Claypool Publishers, 2014.

[9] T. Thelen, "Automatische Analyse orthographischer Fehler bei Einzelwortschreibungen," Master's thesis, Universität Osnabrück, 1998.

[10] K. Berkling, J. Fay, and S. Stüker, "Speech technology-based framework for quantitative analysis of German spelling errors in freely composed children's texts," in *SLaTE*, 2011, pp. 65–68.

[11] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

[12] T. Schmidt and K. Wörner, "EXMARaLDA: Creating, analysing and sharing spoken language corpora for pragmatic research," *Pragmatics*, vol. 19, no. 4, pp. 565–582, 2009.

[13] T. Schmidt, K. Wörner, H. Hedeland, and T. Lehmberg, "New and future developments in EXMARaLDA," in *Multilingual Resources and Multilingual Applications. Proceedings of GSCL Conference 2011 Hamburg.*, T. Schmidt and K. Wörner, Eds., 2011.

[14] U. D. Reichel, "PermA and Balloon: Tools for string alignment and text processing," in *Proc. Interspeech*, Portland, Oregon, 2012.

[15] U. D. Reichel and T. Kisler, "Language-independent grapheme-phoneme conversion and word stress assignment as a web service," in *Elektronische Sprachverarbeitung: Studientexte zur Sprachkommunikation 71*, R. Hoffmann, Ed. TUDpress, 2014, pp. 42–49.

[16] A. Schiller, S. Teufel, C. Stöckert, and C. Thielen, "Guidelines für das Tagging deutscher Textcorpora mit STTS," Universities of Stuttgart and Tübingen, Tech. Rep., 1999.

[17] S. Schroeder, K.-M. Würzner, J. Heister, A. Geyken, and R. Kliegl, "childLex: A lexical database of German read by children," *Behavior research methods*, pp. 1–10, 2014.

[18] G. Thomé, *Orthographieerwerb: Qualitative Fehleranalysen zum Aufbau der orthographischen Kompetenz*. Frankfurt a. M.: Peter Lang, 1999.

[19] J. Fay, *Die Entwicklung der Rechtschreibkompetenz beim Textschreiben: Eine empirische Untersuchung in Klasse 1 bis 4*. Frankfurt a. M.: Peter Lang, 2010.

[20] M. Mangold, *Duden (Band 6). Das Aussprachewörterbuch*, 6th ed. Mannheim: Dudenverlag, 2005.

[21] R. Wiese, *The Phonology of German*. Oxford: Oxford University Press, 2006.

[22] K. Berkling, R. Lavalley, and U. Reichel, "Systematic acquisition of reading and writing: An exploration of structure in didactic elementary texts for German," in *Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology*, Duisburg/Essen, Germany, 2015, pp. 67–76.

[23] J. Reichen, "Lesen durch Schreiben: Lesenlernen ohne Leseunterricht," *Grundschulunterricht Deutsch*, vol. 2, pp. 4–8, 2008.

[24] L. Bebout, "An error analysis of misspellings made by learners of English as a first and as a second language," *Journal of Psycholinguistic Research*, vol. 14, no. 6, pp. 569–593, 1985.

[25] H. Frieg, "Sprachförderung im Regelunterricht der Grundschule: Eine Evaluation der Generativen Textproduktion," Ph.D. dissertation, Ruhr-Universität Bochum, 2014. [Online]. Available: http://www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/FriegHendrike/diss.pdf

[26] P. Eisenberg, "Die Schreibsilbe im Deutschen," in *Schriftsystem und Orthographie*, P. Eisenberg and H. Günther, Eds. Tübingen: Niemeyer, 1989, pp. 57–84.