A connectionist model of Jürgen Reichen's *Lesen durch Schreiben* method

Jannis Born[1], **Nikola Nikolov**[1]

1 *Institute of Neuroinformatics, ETH Zürich and University of Zürich, Switzerland*

Abstract

Within the German language region, the teaching method *Lesen durch Schreiben* (Reichen, 1988) is contemporarily among the most controversially discussed. Despite lacking clear evidence over traditional approaches to teach children writing and reading, it enjoys great popularity among teachers. Empirical investigations of *Lesen durch Schreiben* (LdS) require extensive human resources, are time consuming and costly but yet have difficulties extracting the learning efficacy of the method. Herein, we utilize deep learning and natural language processing techniques to present an encoding-decoding bidirectional recurrent neural network; the first connectionist model that investigates the effect of a LdS inspired training regime. It is evaluated against a classical, "Fibel"-based learner as well as other artificial agents. While this isolates statistical effects that could be induced by LdS on human learners, it does not capture higher cognitive processes influencing the learning progress, but can yet be instructive by shining a more theoretical perspective on the current debate. Our results show that stereotypical spelling mistakes of children exposed to LdS can be replicated with neural network models. These misspellings arise naturally during writing acquisition of all artificial agents, but are either suppressed or reinforced depending on the learning regime. Whilst the LdS learner shows a significant performance reduction in writing tasks, it does so to a much lesser extent in reading. We reveal insights into the network's internal representations and extract biases in the LdS learner that may have induced its increased confound of graphemes with similar phonetic correspondings (such as **<t>** and **<d>**). Overall, we present a modest step towards building a somewhat biologically inspired framework of LdS that may act as a starting point for future work increasing realism on various ends we have not tackled herein.

All the code is available on https://github.com/dopexxx/LdS_bLSTM.

KEYWORDS: Natural language processing, teaching research, connectionism, cognitive science, computational modeling, computational cognitive science, Lesen durch Schreiben, Reichen-method, didactics, education

Introduction

What is Lesen durch Schreiben?

In countries with alphabet-based languages, primary school pupils are usually familiarized with the alphabet on a character-by-character basis. This process is often interblended with reading and writing tasks provided by an alphabet book or primer (in the following in German: *Fibel*) that initially focuses on words that are spelling-to-sound-regular. As an alternative to Fibel-based learning Reichen (1988) conceptualized *Lesen durch Schreiben* (LdS), a teaching method utilizing a phonetic scheme that maps a given phoneme to one or more letters. Part of the LdS framework is to instruct children to write according to their own phonetic decomposition. Children proceed sound-by-sound through spoken words and use the phonetic scheme in order to map a given phoneme to one or more letters. Despite German having a relatively regular orthography compared to e.g. Englisch (Ziegler et al., 2000), this mapping is irregular (e.g. **/v/** vs. **/f/**, letter doubling etc.) resulting in highly defective writings (e.g. **<Varat>** instead of **<Fahrrad>**, i.e. bicycle). Reichen further postulated that correcting the produced mistakes hampers the children's motivation to learn writing. Accordingly, teachers and parents are compelled to not correct orthographic mistakes. This explicitly concerns orthographic mistakes which do not render semantic understanding impossible (i.e. **<Hunt>** instead of **<Hund>** would be accepted whereas **<Kund>** would be corrected). LdS does not include particular reading exercises, but assumes that reading abilities arise automatically based on the own (erroneous) writings, which gives birth to the method's name Lesen *durch* Schreiben (engl. Reading *through* writing). Whilst Reichen has developed a broad didactic framework, the usage of the phonetic scheme and its consequence to tolerate spelling mistakes is most discriminative and therefore subject of the presented work. LdS is widely applied in Germany, Switzerland and Austria, but has yet been controversially discussed over the last decades (Valtin, 1998; Röber-Siekmeyer and Spiekermann, 2000; Lorenz, 2017).

Empirical studies of LdS

A metaanalysis by Funke (2014) showed that a number of empirical studies (conducted between 1985 and 2010) could not consistently reveal a positive or negative effect of the LdS method. Children exposed to LdS seemed to be equipped with slightly weaker writing skills after year 2-4 and exhibited comparable reading skills. As Funke notably point out, the compared studies differ strongly in setup and evaluation techniques and are further sensitive to confounds such as heterogeneous cognitive prerequisites, e.g. due to recent immigration. Conducting further empirical studies to compare effects of LdS to Fibel-based methods comes with various pitfalls. First, these studies are inherently time consuming as they require researchers to monitor classes over the entire 4 years of primary school. Secondly, the extensive human resources involved to select, assess and evaluate these sample sizes require financial resources. Thirdly, Funke (2014) questions to what extent empirical studies can isolate the learning *efficacy* of LdS and emphasizes that previous evidence (Sander, 2006; Friedrich, 2010) rather reported observable learning *results* which, problematically, may to a greater or lesser extent be

induced by the method itself. Funke (2014) concludes that the investigated empirical studies give no reason to infer about the learning efficacy of LdS. As a simple example, assuming that the learning efficacy of LdS is below the one of classical approaches, a LdS-trained children may exhibit a better learning result in a writing task due to intensive private tuition by their skeptical parents.

Scope of the presented work

Using connectionist methods, we present a computational learning agent that can be exposed to a LdS inspired training regime. The dynamics of this training regime are compared to an architecturally identical model that was trained under a Fibel-based training regime in an attempt to answer a question of the following nature:

Is it, from a statistical perspective, advantageous for an artificial learning system to use the *Lesen durch Schreiben* method instead of Fibel-based methods to learn writing and reading?

Importantly, empirical work could not unravel the learning efficacy of LdS compared to Fibel-based methods, but investigated to what extent learning results may be meaningful about or attributable to the learning efficacy of either method (Funke, 2014). Conversely, a simulation bears the advantage to isolate the learning efficacy, but does so only for an artificial learning system, leaving open to what extent comparisons to human learners can be drawn. This being the case, we do not arrogate to reveal anything about suitability of LdS within primary schools, but rather strive to objectify the learning process in order to examine from a learning-theoretical perspective whether it is beneficial to not correct a certain set of mistakes.

By means of established techniques from Natural Language Processing (NLP), we present a self-learning model that consists of two separate modules, one for writing and one for reading. As described in detail in the model architecture, both modules are arranged serially such that the reading learning process bases on the writings generated by the agent. This was done since the Reichen method does not explicitly schedule reading tasks, but assumes the reading abilities to arise naturally as writing abilities elevate. Precisely, a child that mistakenly wrote <**Tso**> rather than <**Zoo**> (engl. zoo) but did not receive any correction from an external teacher will use this very same spelling and try to convert it to the phonetic sequence **/tso:/**. This can be loosely related to the child trying to read its own text to a teacher at a point in time where it has forgotten the semantic content. In our model, the process of writing boils down to a phoneme-to-grapheme conversion (P2G). A child willing to write the word <**elephant**> needs to have its phonetic representation available at first. Reading, instead, is the inverse process, a grapheme-to-phoneme conversion (G2P). The orthographic information is available from an external source, but the challenge is to decompose the sequence of letters into smaller units that each correspond to a particular phoneme.

The model can be trained by two regimes, (1) a traditional, Fibel-inspired learning regime (called *regular*) in which all orthographic mistakes are corrected by a teacher or (2) a LdS-inspired learning regime in which only those orthographic mistakes are corrected that result in a different sequence of phonemes when spoken out by a teacher. Simply put, a children mistakenly spelling <**Zoo**> as <**Zo**> or <**Zoh**> or <**Tsoh**> would be penalized in the former but not in the latter regime. Instead, spelling it as <**Ze**> would still be corrected in identical fashion within both regimes. Over the course of training, the regimes are combined in different ways to yield learning agents that make certain assumptions about the nature of stimuli (for details see Training regimes).

Importantly, there are plenty components of both, the LdS framewowk and reading/writing acquisition in general that are not tested in this work; non-exhaustively, but including:

1. **Frustration.** It is often brought forward by LdS advocates that extensive correction by teachers induces frustration, a lack of self-confidence and takes joy out of learning, which ultimately impaires the learning result. There is no qualitative equivalent for this notion herein.
2. **Neuronal populations.** Neurophysiologically, many areas involved in reading also play a role in writing tasks, such as fusiform gyrus, inferior frontal gyrus and Broca's area (for a detailed review see Tsapkini and Hillis (2013)), whereas our model solves both tasks with completely distinct neuronal populations.
3. **Spelling-to-sound-regularity.** A Fibel usually starts with sound-to-spelling-regular words such as <**Mama**> whereas irregular words are introduced later; a fact that has not yet been incorporated into our model.
4. **Motoric skills** that are required to produce readable letters and understandable sounds.
5. **Immigrated children** that speak predominantly a language other than German may associate different phonemes to the pictograms on the phonetic scheme. Instead, an assumption our model makes is that during writing the agent can completely access the phonetic transcript of the desired written word, i.e. it can speak out the word perfectly.

Since Seidenberg and McClelland (1989), connectionist models of language have been utilized to understand the acquisition of human language. Initially, they challenged the most accepted representational account of reading and spelling, the dual-route model (Coltheart, 1978; Coltheart et al., 2001). It suggests that reading proceeds through two disjoint tracks, whereby familiar words are processed through a direct, lexical route and unfamiliar words through a sublexical, phoneme/grapheme-wise route. Later, Plaut et al. (1996) argued that their connectionist model, nowadays known as triangle model of reading, was indeed a three-route model, despite it uses the same computational units for pronouncing every word. In another work, Brown and Loosemore (1995) showed that artificial agents tend to make similar mistakes in reading tasks like both, normal and dyslexic children.

In this work, both reading and writing module perform a stepwise graphological/phonological recoding, corresponding more to a sublexical/indirect route. As an upgrade in comparison to older connectionist approaches (Seidenberg and McClelland, 1989; Plaut et al., 1996; Hutzler et al., 2004) its bidirectional nature enables a holistic perception of each word by proceeding forwards and backwards through a word before pronunciation/spelling starts. Arguably, this could represent a lexical access, however, it is by no means faster than the stepwise recoding but rather a result of simultaneous forward and backward

recoding. This very same process is used for all words. The dataset is retrieved from the prevalent CELEX corpus (Baayen et al., 1995) that e.g. Hutzler et al. (2004) used to compare the pace of reading skill acquisition in more or less regular orthographies (German and English) between artificial and human learners.

Methods

Framework

This subsection gives a modest overview about recent advancements in sequence learning that served as an inspiration for the architecture of the presented model.

As mentioned above, we treat writing as a P2G conversion task and reading as the inverse problem (G2P). Remarkably, P2G and G2P model architecture can be used interchangeably by swapping the input/output data streams (Shaik, 2016). This is due to the fact that both graphemes and phonemes come in sequences. Since the great progress in NLP within the last decade was particularly based upon recent success of deep learning (Young et al., 2017), we put aside mode traditional approaches of the connectionist framework (such as MLP) and built upon the gold standard for sequence-to-sequence learning, namely the encoding-decoding scheme that utilizes two recurrent neural networks (Cho et al., 2014; Sutskever et al., 2014). Recurrent neural network (RNN) can be considered as a generalization of standard feedforward neural networks to sequences. A RNN computes a sequence of outputs $(y_1, ..., y_n)$ as response to a sequence of inputs $(x_1, ..., x_n)$ by iteratively updating its recurrent state equations. Problematically, most NLP challenges such as language translation or end-to-end automatic speech recognition accompany input and output sequences of different and varying lengths[1]. The encoding-decoding scheme solves this problem by utilizing a first RNN to map the length-varying input sequence to a fixed-sized vector (usually of much higher dimensionality than the input sequence). Next, a second RNN maps this fixed-sized vector to the length-varying target sequence. Within the encoding-decoding scheme, the RNN units can be anything from vanilla RNN, LSTM (like in Sutskever et al. (2014)) or the recently introduced Gated Recurrent Unit (GRU, Cho et al. (2014)). Since vanilla RNN units are known to have difficulties learning long-range temporal dependencies due to vanishing/exploding gradients introduced by Backpropagation through time (BPTT) unrolling the network into a deep feedforward network (Hochreiter et al., 2001), Long Short-Term Memory (LSTM) units are usually used instead. For conceptual details of LSTMs see Hochreiter and Schmidhuber (1997); Olah (2015). For a recent analysis of their pitfalls, ways to regularize them and the implementation used in this work, see Zaremba et al. (2014). A well-known problem is that RNNs only consider *previous* but not *future* contextual dependencies since they process input in temporal order. For example a LSTM used for Part-of-Speech Tagging will frequently suffer from the garden-path-effect such as recognizing the final verb phrase in "The horse raced past the barn fell". Bidirectional RNNs (bRNN), first introduced by Schuster and Paliwal (1997) are a dexterous extension overcoming this shortfall by presenting the input sequence in both regular and reversed order. Conceptually, bRNNs enable the learning agent to move forward and backward through the input sequence before a decision on the outputs is made, somewhat comparable to a child repeatedly speaking out a word loudly before it writes it down. An instance of bRNN, bidirectional LSTMs (bLSTM) were introduced later for speech recognition tasks such as phoneme classification (Graves and Schmidhuber, 2005; Graves et al., 2013). Quickly after the release of the encoding-decoding scheme for sequence learning in 2014, bidirectional RNNs such as a bidirectional GRU were utilized within the encoder (Bahdanau et al., 2014). For both the G2P and the P2G module, we implemented the encoding-decoding scheme with a layer-stacking of LSTM units in the encoder, similar like in Luong et al. (2015), but extended by bidirectionality whilst omitting the attentional mechanism

Model architecture

Following a top-down approach, this subsection describes the architecture of the presented model in detail.

Each learning agent was constituted of two separate modules, one for writing and one for reading, both having identical architecture.

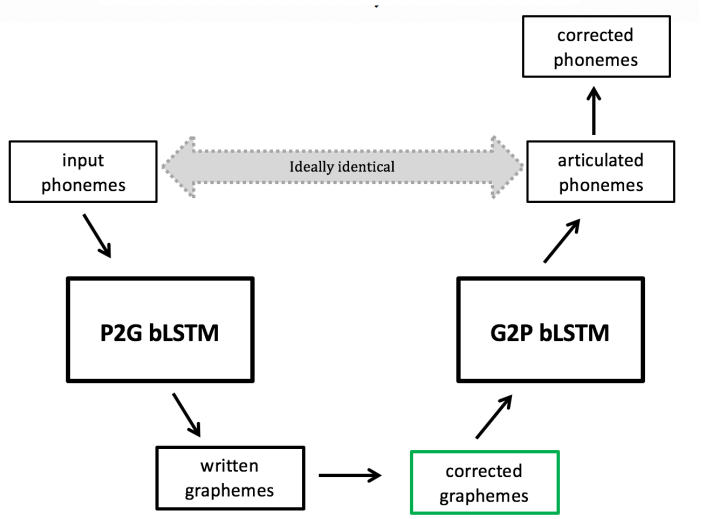

Figure 1: **Model of the learning agent.** The left part displays the writing module, converting phonetic into graphemic sequences, whereas the reading module (right side) does the converse operation. If both these independent operations were successful, input and articulated phonemes are identical. The green box shows the only position where the regular learning agent differed from the LdS agent.

Figure 1 shows that these modules were serialized such that each training step started with a writing task, followed by a reading task of the exact same word. Initially, the writing (P2G) module is shown an input phoneme sequence such as **/hʊnt/** that can loosely be interpreted as a concept popping up in a child's mind when it wants to write a word. The P2G bLSTM then transforms the phonetic into a graphemic sequence (details on the architecture are given further below). This process will initially be random, after some learning highly erroneous and after more learning ideally perfect. Within each training step, the output sequence ("written graphemes"), e.g. <**Hunt**> will be corrected according to the current learning mechanism (for details see loss function). Crucially, in the next step, the possibly

1. Apparently, the same model that translates **/tso:/** into <**Zoo**> should be able to transfer **/ˈʁʊkˌzak/** into <**Rucksack**>

corrected graphemic sequence is fed to the G2P bLSTM that tries to convert it into a phonetic sequence. In case both steps are accurate, the output of the reading module is identical to the input of the writing module; otherwise a correction mechanism, this time identical across for both learning regimes, will again be applied. This macro-architecture bears slight similarity to the autoencoder setup, however, here (1) both components are separate entities trained individually and supervised with gradient descent and (2) the latent variable (written/corrected graphemes) is not intended to be condensed with respect to the input. The serialization of writing and reading module is a natural way of dealing the fact that LdS does not explicitly schedule reading tasks from books but assumes reading abilities to arise inherently by recapitulating the own writings. As a side note, the reading module could not "memorize" the input phoneme sequence fed to the writing module since it was a computationally separate unit.

The time-unrolled architecture of the reading module of Figure 1 is shown in Figure 2.

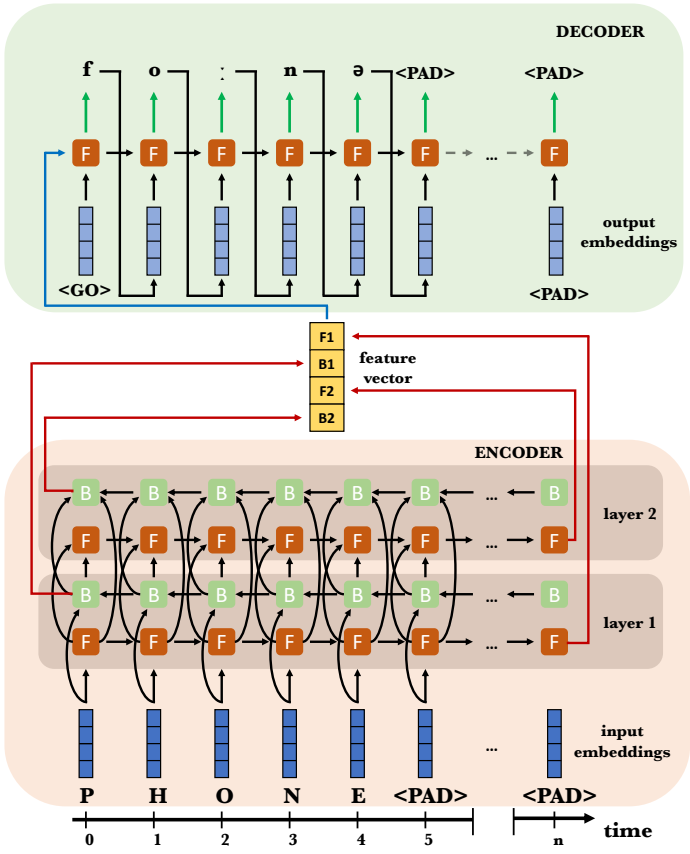

Figure 2: **G2P module architecture.** Time-unrolled architecture of the encoding-decoding bLSTMs as shown in Figure 1. Orange nodes indicate forward, green nodes backward LSTM cells. Black arrows represent data being fed into the receiving node, red arrows indicate a concatenation operation (shown for illustration purposes), the blue arrow indicates hidden state initialization and the green arrows indicate two fully connected layers used to map the encoder LSTM's output to the output vocabulary. For details, please refer to the text.

For every time step, one letter of the input sequence is given to the encoder. First, the current letter's corresponding embedding vector (96 dimensions) is retrieved from the embedding lookup matrix. Next, this embedding is passed separately to both units in the first bLSTM layer. This layer consists of one LSTM cell that receives the input sequence in forwards order (orange F nodes) and one receiving the input in backwards order (green B nodes). After the hidden states and output states of both LSTM cells are computed, their outputs are given to both, forward and backward LSTM of the subsequent layer. The dimensionality of the hidden and output states of all LSTM cells was set to 128 across all simulations. When the entire input sequence has been processed, the hidden states of the last time step of the two forward LSTMs are concatenated with the hidden states of the first time step of the two backward LSTMs, yielding a 512-dimensional feature vector representation of the input sequence (shown in yellow). The decoder consists of a single forward LSTM cell that is initialized by this feature vector like in Cho et al. (2014). Subsequently, the LSTM is updated by the output embedding vector (again of length 96) of a special <GO> character which instructs the LSTM to start outputting the target sequence (phonetic in this example). Its 512-dimensional output is then classified to the predicted phoneme by two densely connected feed-forward layers with 128 units in the hidden layer and a *linear* activation function (for simplicity not shown in detail). The next step differs between training and testing phases: During testing, in order to generate the next target token, the output embedding vector of the predicted token of the current time step is used as an input for the LSTM cell of the next time step. During the course of training, however, the decoder setup differed as shown in Figure 3.

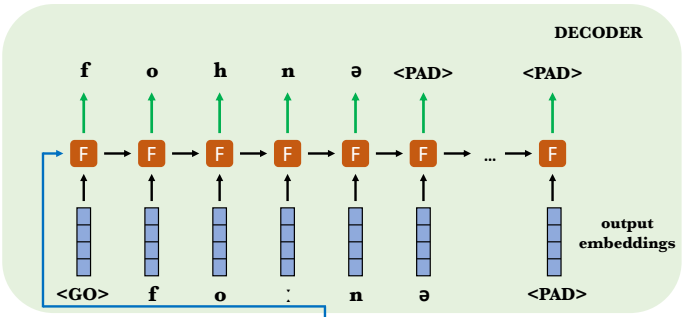

Figure 3: **Encoder during training.** Time-unrolled architecture of the encoding-decoding bLSTM during training. The true labels, offset by an index of 1, were used as inputs of the encoder. Legend like in Figure 2.

Instead of feeding the model's predicted output as input for the next state of the LSTM cell, the actual target vector, shifted by one element is used. For example at $t = 2$, the model in Figure 3 produces the wrong stretching symbol (**/h/** instead of **/ː/**). However, assuming the model had correctly guessed the previous token, the true label **/ː/** was used for the next LSTM cell update. This trick, introduced by Bengio et al. (2015), is a standard procedure during the training regime of sequence learning models and it is deemed to help the model dealing better with its own mistakes (otherwise, early mistakes would be amplified such that the model prediction quickly deviates away from the actual targets). In order to regularize this fairly complex model, a rather high dropout probability of 0.5 was applied during training. Following Zaremba et al. (2014), dropout was not applied on the recurrent connectivity of the LSTM cells, but on its inputs (no dropout on the outputs). The dropout mask was recomputed at every time step, making it unlikely but possible that instead of the input embeddings only zero-rows were fed into the model. Dropout was also applied on the first fully connected layer in the decoder but not directly to the phonetic and orthographic embeddings. Input and output sequences were

padded in practice in order to enable variable-length computations.

The writing module had exactly the same structure like presented in Figure 2 and Figure 3 (the example would need to be inverted though) and both these modules were used in the LdS and the regular agent. In other words, the forward pass was identical across all compared tested agents, the LdS agent differed from the regular agent only by the applied training regime (i.e. the loss function). The model was implemented in Python 3.6 and Tensorflow 1.7.0 and trained on a GTX 1080 GPU.

Loss function and learning agents

SEQUENCE LOSS FUNCTION MODIFICATION

Both, during the LdS and the regular learning regimes stochastic gradient descent implemented through the ADAM optimizer was used to train the model (Kingma and Ba, 2014). The loss function subject to minimization is a weighted cross-entropy loss for a sequence of logits which was chosen over the CTC loss since it can produce output sequences longer than the input sequence (Graves et al., 2006; Hannun, 2017). No gradient clipping was applied.

For the writing module of an agent in the LdS training regime, a biased variant of the weighted cross-entropy loss function was created according to the following perturbations:

Prior to training, a list of alternative, "correct" (in the LdS sense) spellings was generated by a mechanism described below (e.g. [<**zoo**>,<**zo**>,<**tso**>,<**tsoh**>, ...]. During training, this list was used as a lookup table to verify whether the spelling produced by the agent occurred in the list of alternative writings. If this held true, the corresponding spelling replaced the actual target sequence before the weighted cross-entropy loss was computed regularly. By means of that, the actual logits, carrying confidence values of the occurrence of every letter from the vocabulary at a given time, were left unaltered, whereas only the teacher signal was amended. In other words, the agent's certainty about its just outputted spelling remained the same, just as it is before the teacher signal is applied. We refer to this biased loss function as *LdS training regime*.

TESTED LEARNING AGENTS

Figure 4 shows how the LdS and the regular training regime were combined over the course of training in order to investigate five different learning agents:

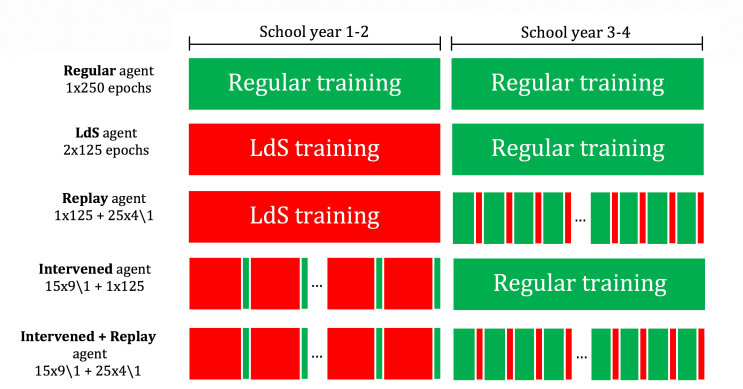

Figure 4: **Training regimes.** The regular, Fibel-inspired, training regime shown in green and the LdS inspired training regime (red) were combined in 5 different ways in order to create learning agents that made different assumptions about the real world. The abscissa shows the temporal evolution during training. Details in the text.

1. **Regular agent:** This agent is thought to approximate a child trained writing and reading through classical, Fibel-based approaches. It makes use of the regular training regime throughout the entire course of training. All its mistakes done on writing tasks are corrected instantaneously by a teacher and all graphemic sequences used as inputs for the reading module have correct spellings.

2. **LdS agent:** This agent is the naive approach to approximate a child taught via LdS. During the first two years of school (the first half of training), it is exclusively trained via the LdS regime, in which phonetically (almost) indistinguishable spellings are actively reinforced by a teacher. During this period, the reading module of the LdS agent may fed with erroneous non-words such as <**tsoh**> from which it tries to retrieve **/tso:/**. When the second half of training begins, the learning regime changes to regular, i.e. the agent is suddenly confronted with the situation that previously encouraged spellings get penalized.

3. **Replay agent:** According to the Complementary Learning Systems (CLS) framework the interplay of the hippocampus as sparse, rapid learner of episodic memories and the neocortex as dense extractor of latent semantics enables continual learning (i.e. to overcoming catastrophic interference, McClelland et al. (1995)). Crucial in this reciprocity is hippocampal replay, a process of memory consolidation that interleaves salient, recalled memories with contemporary experience (O'Neill et al., 2010; Kumaran et al., 2016). Hence, it has to be contemplated that children in the regular training regime recall past episodes where incorrect spellings of a word were accepted by a teacher. The replay agent implements this by interleaving 4 epochs of the regular regime with one LdS epoch during the second half of training.

4. **Intervene agent:** Conversely, parents of children exposed to LdS training may intervene the learning process in an attempt to teach the correct spelling of a word. The intervened agent thus alternates 9 epochs of LdS training with one regular epoch during the first half of training.

5. **Intervene + Replay agent:** This agent combines the assumptions of the Intervened and the Replay agent.

Datasets

ChildLex

In an attempt to train the model on words that realistically capture the vocabulary of a primary-school pupil, we extracted 10000 lemmata that the ChildLex database attributes to 6-8 year-old children (Schroeder et al., 2015). ChildLex comes without phonetic representations, thus the phonetic representations were retrieved from the second release of the CELEX database instead (Baayen et al., 1995). Foreign words with the phonemes **/æ/, /ɑ/, /ʒ/** or **/ʧ/** were excluded, whereas homophones were included. All words were converted to lowercase. After excluding words with more than 50 000 alternative writings (procedure described below), 5891 remained in the corpus.

CELEX

As universal function approximators, neural networks tend to have low bias but high variance (i.e. risk of overfitting), in particular if dataset size is too small (Guan and Plötz, 2017). We therefore tested the model also on all samples from the CELEX database. Since in the copy we were offered, only a small fraction came along with phonetic transcripts (ca. 50 000 out of

517 000) we used the German version of the text-to-speech-synthesizer espeak to generate the phonetic transcripts. We applied the same exclusion criteria like above, but for all words with > 100 alternative spellings, we added a randomly sampled subset of size 100 to the corpus, leading to a total of 416 495 samples; a dataset size more suited for the expressiveness of the model.

Generation of alternative spellings

Ethos of LdS is that the qualitative understanding of a native speaker reading the text written by a child has a greater value than actual graphemic correctness, i.e. for every word multiple spellings are accepted by a teacher. To our best knowledge, neither Reichen (1988) nor any of the more recent sources has explicitly defined rules on how to find the set of spellings that are acceptable for a given word. In order to generate the alternative spellings for all words of the corpora, a set of 46 rules mapping IPA symbols on letter sequences was defined (for details see Appendix). In brief, like researchers do it in evaluation of spelling pseudo-words, all spellings that represented the phonetic sequence accurately were accepted (Klicpera et al., 2013). The algorithm was as follows:

Algorithm 1 Generate alternative spellings

```
alt_spellings ← [ ]
for every phonetic sequence ps in corpus do
    {%Find where and which alternative letters are ok}
    ipa_list ← split ps into list of IPA symbols
    possible_tokens ← [ ]
    num_alt_spellings_ps ← 1
    for every ipa_symbol i in ipa_list do
        g ← set of possible graphemic transcriptions of i
        possible_tokens.append(g)
        num_alt_spellings_ps ← num_alt_spellings_ps ·|g|
    end for

    {%Find all combinations of allowed letters}
    alt_spellings_ps ← [ ]
    i ← 0
    while i < num_alt_spellings_ps do
        spelling ← [ ]
        for every set s in possible_tokens do
            gs ← exactly one element from s
            spelling.append(s)
        end for
        if spelling not in alt_spellings_ps then
            alt_spellings_ps.append(spelling)
            i ← i + 1
        end if
    end while
    alt_spellings.append(alt_spellings_ps)
end for
```

This automatized procedure was favored over a hand-crafted approach since the problem suffers from combinatorial explosion. For example, a simple phonetic sequence like **/reːgaːl/** (<**Regal**>, i.e. shelf) consists of the five phonetic units **/r/**, **/eː/**, **/g/**, **/aː/** and **/l/** corresponding, according to Table 1, to respectively 3, 3, 3, 3 and 2 possible spellings, resulting in a total of 162 spellings (some of them as absurd as <**Rheeghahll**>). This procedure was much faster than creating alternative writings by hand, but produced, for some phonetic sequences, as many as 35 million alternative writings. Whilst this was an instructive lesson regarding the freedom LdS children have in their spelling, it forced us to cut off all words from the corpus that had more alternative writings than some threshold.

Hyperparameter

We are well aware that utilizing datasets of varying sizes in principle requires separate tuning of network size (depth, neurons per layer, embedding dimensionalities etc.). As we were predominantly interested in relative comparisons between different models rather than absolute performance, this has been discarded for economical reasons. In all cases, 5% of the dataset was hold back for testing. Batch size was 20 and 2000 respectively. The model performance on train and test dataset was evaluated every epoch (without dropout).

Model analysis

Model performance for both writing and reading module was assessed via plain token and word accuracy. Token accuracy was measured by the Levensthein distance (Levenshtein, 1966), a string similarity metric that is commonly used to measure orthographic neighborhood by counting the minimal amount of insertions, deletions and substitutions necessary to warp the predicted sequence $\hat{\boldsymbol{y}}$ into the target sequence $\boldsymbol{y}$ (Yarkoni et al., 2008). Word accuracy is the ratio of samples that had $L(\vec{\boldsymbol{y}}, \vec{\hat{\boldsymbol{y}}}) = 0$.

Further, the nature of mistakes made by both agents was systematically analyzed and brought in accordance with perturbations in the embedding vector representations visualized by dimensionality reduction techniques such as PCA or t-SNE (Maaten and Hinton, 2008).

Results

ChildLex corpus

Figure 5 compares different performance metrics of all five investigated learning agents on the ChildLex corpus. As the plot on the upper left suggests, the agent trained under the LdS regime had a significantly lowered word accuracy (84% vs 90%) during the first half of training. This shows that the exposure of the LdS agent to a biased training objective indeed induced suboptimal writings. The LdS agent consistently misspelled some words that were correctly learned by the regular agent.

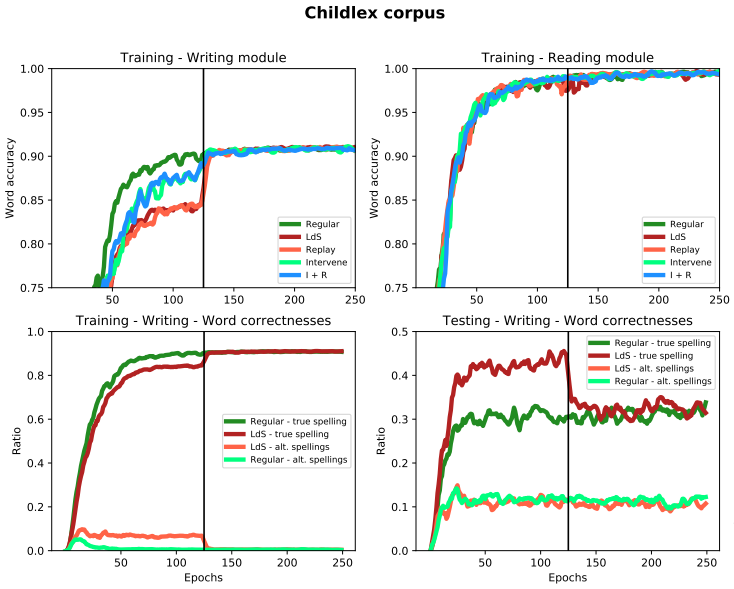

Figure 5: **Development of word accuracy on the ChildLex corpus over the course of training.** *Upper left:* Writing performance of all five agents on training data. *Upper right:* Their reading performance on training data. *Lower left:* Ratio of correct spellings and ratio of LdS-accepted spellings for regular and LdS agent on training data. *Lower right:* Same plot for testing data. The vertical bar after 125 epochs indicate the change in the learning regime for the LdS agents. All lines were smoothed with moving window of size 5 for visual clarity.

Common types of errors where confounds of (1) <**d**> and <**t**> (<**Walt**> for <**Wald**> or <**Wird**> for <**Wirt**>), (2) the German "Dehnungs-h" (<**fro**> for <**froh**>) or (3) consonant doublings (<**derr**> for <**der**>). However, rapidly after the change of the regime, the LdS agent catches up to the Regular agent, which can be associated to the known phenomenon of catastrophic interference, i.e. an agent that learns and masters task A and then learns another task B, will forget the knowledge about task A (French, 1999). Next, the parents' tutoring in the intervene agent had a positive effect on its writing abilities. Despite only every 10th epoch was a regular rather than a LdS epoch, the intervene agent closed more than half of the gap to the regular agent. This suggests that even sporadic correction can nudge the agent to the correct spelling, since it would also be accepted from the teacher during the LdS regime. Similar tendencies were observed for token accuracy of the training data. Interestingly, as the plot on the upper right shows, the LdS training regime does not impair the reading performance. This may be due to the fact that for the reading module, not the learning mechanism, but only the input data was altered. Indeed, this data augmentation, a common technique to improve performance in neural networks (Cireşan et al., 2010), lead the LdS agent outperform the Regular agent on the test data (not shown). The plot on the lower right takes a closer look at the Regular and the LdS agent by comparing how many words they spelled correctly (upper, darker lines) or correctly from the perspective of a LdS teacher (lower, brighter lines). As the early rise in the light green line indicates, the Regular agent also produced LdS-accepted spellings, albeit they rapidly got suppressed by the teacher's correction. However, due to the limited size of the ChildLex corpus this suppression/reward did not generalize to the test data (bottom right of Figure 5) where both agents produced equal amounts of "LdS-correct" spellings. Surprisingly, the LdS agent surpassed the rRegular agent on the test dataset, which was probably caused by the LdS agent overfitting less due to the data augmentation by the amended correction mechanism. The gap in absolute accuracy between training and testing is also an apparent sign of overfitting which led us to investigate model performance on a larger corpus.

CELEX corpus

Figure 6 shows the results after the model was trained on the CELEX corpus.

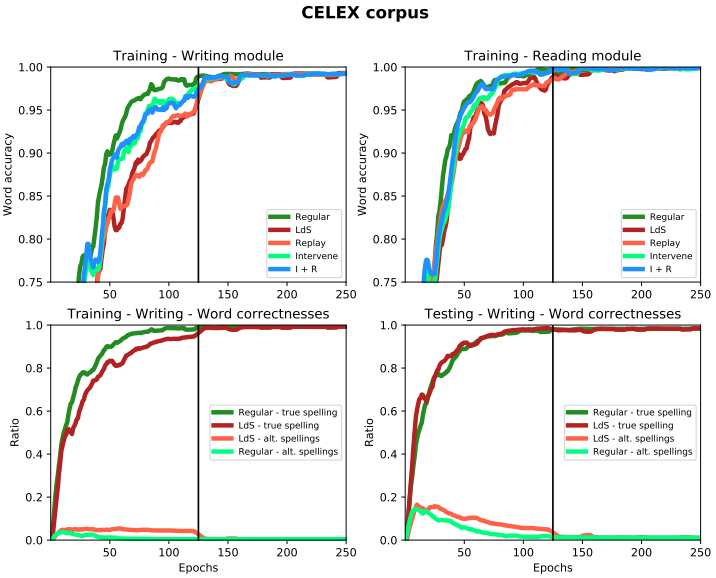

Figure 6: **Development of word accuracy on the CELEX corpus over the course of training.** Same legends like in Figure 5. All lines were smoothed with moving window of size 9 for visual clarity.

In contrast to the ChildLex corpus, this enlarged dataset prevented the model from overfitting. Training and testing performances were almost identical, both reached nearly 100%, with the Regular agent having only 301 spelling mistakes and 114 pronunciation mistakes out of 20 825 testing samples (accuracies of 98.6 and 99.5% respectively). Similarly to the ChildLex corpus, the LdS and the Replay agent, which were exposed exclusively to the LdS regime during the first half of training, had a lowered word accuracy in spelling (ca. 95% vs 99%), whereas intervening the LdS regime with regular epochs elevated writing performance to a certain extent (upper left plot). As the upper right plot suggests, also the reading performance of the LdS was impaired. This time, the dataset was large enough to demand the model to expend its entire flexibility to capture the variance in the data. Therefore, the additional perturbation induced by the LdS agent reading its own erroneous writings, hampered the reading performance. When the reading module was presented the erroneous writings, it still had not fully extracted the information from the correct training data, but since the LdS-biased samplings were presented, it was kept away from recapitulating the correct spellings. Whilst the bottom left plot of Figure 6 again suggests that the LdS agent learned suboptimal spellings that were suppressed for the regular agent, the bottom right plot reveals a decrease in LdS-accepted spellings for *both* the regular and the LdS agent. We suspect this to arise from the fact that our procedure to generate the alternative writings produced more than 100 alternative writings for the vast majority of the words.

However, for economical and computational reasons, when a word had 12 000 alternative writings, we defined the alternative writings that the teacher would accept as a random subset of size 1000. Apparently, this destroyed regularity and structure in the alternative writings and to the LdS agent it occurred that stereotypical mistakes (such as forgetting the "Dehnungs-h") were tolerated for some, but not for all words. In other words, the spelling variations the LdS agent learned on the training data did not generalize to the test data which presumably caused the decline in the LdS spelling ratio by the LdS agent over time.

Investigating more closely the nature of mistakes (see Figure 7) revealed that reading was solved almost perfectly: the biggest group of mistakes came from defective usage of the intonation symbols /ˌ/ and /ˈ/.

	Regular agent			
	Writing		Reading	
Type	Vowel type	35%	Intonation	30%
	Consonant type	15%	Missing final consonant	20%
	Homophones	10%	Vowel type	15%
	Consonant doubling	10%	Phoneme insertion	15%

Figure 7: **Stereotypical mistakes.** Most common groups of mistakes the Regular agent did in spelling and pronunciation after 125 epochs of training. Evaluated on training data.

For writing instead, vowel confusion was the most frequent source of error. Also empirically, vowel confusion was reported to be a common source of error (Wien, 2013). 10% of the errors were constituted by writing homophones (e.g. the agent wrote <**das**> instead of <**dass**>), a type of mistake that is virtually impossible to avoid as it requires the context of multiple words to resolve the ambiguity.

Like shown in Figure 8, the LdS agent committed significantly more vowel mistakes, predominantly confounding <**e**> and <**ä**>.

	LdS agent (additional mistakes)	
	Writing	
Type	<e> vs. <ä>	45%
	<d> vs. <t>	10%
	Consonant doubling	9%
	"Dehnungs-h"	9%
	<v> vs. <w>	8%

Figure 8: **LdS agent mistakes** Most common groups of spelling mistakes the LdS agent did *in addition* to the Regular agent after 125 epochs of training. Evaluated on training data.

Moreover, also irregularities in consonant doubling (such as the double <**ss**> in <**Grass**>, but the single <**s**> in <**Gas**>), graphemes with similar, or overlapping pronunciation or the classical "Dehnungs-h" were classic mistakes the LdS agent did, but the Regular agent had overcome. All these mistakes were reinforced by the LdS learning regime that accepted these spellings.

In an attempt to unravel the internal representations of our encoding-decoding bLSTM neural network model, Figure 9 displays a 2D projection of the grapheme representations.

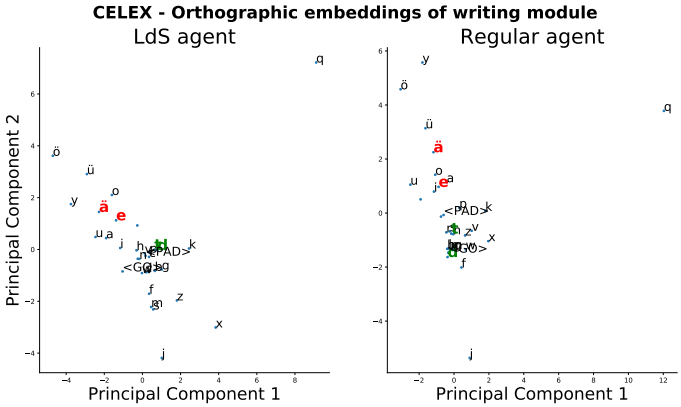

Figure 9: **Learned grapheme representations.** Comparing the PCA projections of the orthographic vectors for the LdS and the regular agent. Perturbations in these representations correlate with tendencies for certain types of errors.

This idea traces back to Osgood (1952), but whilst initially hand-crafted features were used, in recent connectionist models, the features corresponding to every grapheme/phoneme (in our case points in $\mathbb{R}^{96}$) are learned during training. These high-dimensional internal representations can be visualized in 2D by dimensionality reduction methods like Principal Component Analysis (PCA) or t-SNE. Crucially, proximity between two tokens in the low-dimensional space indicates that the agent had a similar, semantic understanding of these tokens. We borrow this technique from language translation models that utilize words as tokens; rather than phonemes or graphemes (Sutskever et al., 2014; Cho et al., 2014). The PCA visualizations in Figure 9 reveal that, for both the Regular and the LdS agent, (1) vowels form clusters and (2) letters that are used irregularly (such as <q>) induce quite independent representations. For an easier comparison of the two learning agents, we have colored exemplary <**e**> and <**ä**> in red and <**d**> and <**t**> in green, since confounds of these two were the most common types of errors that distinguished the LdS from the regular agent. As Figure 9 suggests, both these pairs of letters are in closer spatial proximity for the LdS agent compared to the Regular agent (28% and 83% reduced distance). This may suggest that conceptually the LdS agent had a less clearer distinction of <**e**> to <**ä**> and <**d**> to <**t**> than the Regular agent, which could have induced the increased error rate on these types of words. We have chosen this lossy, compressed two-dimensional analysis for visual clarity, however the described tendency withstands the original high-dimensional representation.

Discussion

To our best knowledge, we have presented the first computational model of a LdS inspired training regime. Comparing a LdS learner to a "Fibel"-inspired learner, we have demonstrated that, within our model, exposure to LdS leads to a significantly decreased performance in writing familiar words. In brief, it is not beneficial but rather harmful for the learning result of an artificial agent to not correct certain types of mistakes. Instead, reading of familiar words is accomplished at the same or almost the same level. As soon as the LdS learner switches to the regular regime (like it usually is done for primary schoolers that enter year 3), it quickly catches up to its regular competitor. This is likely to be caused by catastrophic interference of previously learned task objectives, a longstanding but mostly unsolved shortfall of neural networks Kirkpatrick et al. (2017). In an

attempt to overcome this shortfall, Flesch et al. (2018) recently drew inspiration from human agents to demonstrate that interference can be significantly diminished by compelling an artificial agent to represent incoming stimuli in a factorized (decorrelated) manner. The aforementioned effects were repeated for corpora based on the well-known ChildLex (Schroeder et al., 2015) and CELEX (Baayen et al., 1995) databases. However, the comparably small ChildLex corpus induced overfitting in our model which led us to primarily analyse performance on the CELEX database. Most of the types of mistakes our model committed on the CELEX corpus (vowel type, homophones) are also frequent sources of error in human learners. In addition, shining light on the internal representation of graphemes and comparing the LdS and the Regular agent we report a correlation between error-proneness to confound certain letters and the Euclidean distance between the affected representations. Despite we want to stress that we do not claim a causality to be present, we believe that these representations could be a modest step towards understanding inference in connectionist models. Overall, the reading module outperformed the writing module in its accuracy on both datasets. Since in German, the grapheme-to-phoneme mapping has a greater regularity like the converse phoneme-grapheme mapping this was in alignment with our expectations (Klicpera et al., 2013). Training with CELEX was furthermore an attempt to expose the model to a greater variety of words, in particular to all flections of a lexeme rather than only its lemma. This came at the cost of randomly selecting alternative writings rather than excluding words with > 100 alternative writings. Our uncontrolled random selection process is likely to have destroyed the structured generation process of alternative writings. In other words, if the LdS agent started to over-utilize the German "Dehnungs-h" (stretching-h), its unsystematic acceptance/rejection induced by the incomplete set of alternative writings presumably suppressed the erroneous tendency. Future work should replace our random selection by a more informed process or increase computational power to allow more alternative writings. For the three tested intermediate agents (Replay, Intervene, Intervene + Replay), the performance inclined to be linearly distributed between the LdS and the regular agent during the first of half of training in the sense that more exposure to the regular training regime boosted writing performance. During the second half of training, the Replay and the Intervene + Repl ay agent did not show significant differences from the LdS agent that did not recapitulate past memories of the LdS regime. Critically, our Replay agent was implemented by interleaving regular with LdS epochs, so that during LdS epochs both, correct as well as alternative writings were accepted. This may have been of negligible effect in case the Replay agent had already overwritten the initially alternatively spelled words with the correct spelling during the regular regime. Instead, we should have defined a separate *Replay regime* wherein a teacher accepts the words written with alternative spellings during the LdS regime, only with that particular alternative writing and *not* with the actually correct spelling. Conclusively we think that it can be instructive to take a glance at how artificial learners deal with situations we expose our children to. But in order to capture more realistically the consequences of *Lesen durch Schreiben*, many amendments, some of which are suggested above, need to be done to our framework.

Appendix

IPA → letter mappings. The list of rules used for the generation of alternative spellings is shown in Table 1. It bases on phoneme-grapheme rules in Valtin (2000); Linke et al. (2004); Jaeger (2010)

Phoneme(s)	Grapheme(s)
/a/	⟨a⟩
/aː/	⟨a⟩, ⟨aa⟩, ⟨ah⟩
/aɪ/	⟨ai⟩, ⟨ei⟩
/aʊ/	⟨au⟩
/b/	⟨b⟩, ⟨b⟩
/d/	⟨d⟩, ⟨dd⟩
/ɛ/	⟨e⟩, ⟨ä⟩
/ɛː/	⟨ä⟩, ⟨äh⟩
/e/	⟨e⟩
/eː/	⟨e⟩, ⟨ee⟩, ⟨eh⟩
/f/	⟨f⟩, ⟨v⟩, ⟨ff⟩, ⟨ph⟩, ⟨th⟩
/l/	⟨l⟩, ⟨ll⟩
/g/	⟨g⟩, ⟨gg⟩, ⟨gh⟩
/h/	⟨h⟩
/ɪ/	⟨i⟩, ⟨ie⟩
/i/	⟨i⟩
/iː/	⟨ie⟩, ⟨i⟩, ⟨ih⟩, ⟨ieh⟩
/j/	⟨j⟩
/k/	⟨k⟩, ⟨ck⟩, ⟨c⟩, ⟨g⟩, ⟨ch⟩
/ks/	⟨chs⟩, ⟨x⟩, ⟨ks⟩, ⟨cks⟩, ⟨gs⟩
/kv/	⟨qu⟩
/m/	⟨m⟩, ⟨mm⟩
/n/	⟨n⟩, ⟨nn⟩
/ŋ/	⟨ng⟩, ⟨n⟩
/ɔ/	⟨o⟩
/ɔy/	⟨eu⟩, ⟨äu⟩
/œ/	⟨ö⟩
/oː/	⟨o⟩, ⟨oh⟩, ⟨oo⟩
/øː/	⟨ö⟩, ⟨öh⟩
/p/	⟨p⟩, ⟨b⟩, ⟨pp⟩
/pf/	⟨pf⟩
/r/	⟨r⟩, ⟨rr⟩, ⟨rh⟩
/ʃ/	⟨sch⟩, ⟨s⟩
/s/	⟨s⟩, ⟨ss⟩
/t/	⟨t⟩, ⟨d⟩, ⟨tt⟩, ⟨dt⟩, ⟨th⟩
/ts/	⟨z⟩, ⟨tz⟩
/ʊ/	⟨u⟩
/u/	⟨u⟩
/uː/	⟨u⟩, ⟨uh⟩
/v/	⟨w⟩, ⟨v⟩
/x/	⟨ch⟩
/y/	⟨y⟩
/ʏ/	⟨ü⟩, ⟨u⟩, ⟨y⟩
/yː/	⟨ü⟩, ⟨üh⟩, ⟨tt⟩, ⟨dt⟩, ⟨th⟩
/z/	⟨s⟩
/ˈ/	⟨ ⟩
/ː/	⟨ ⟩

Table 1: Rule list for generation of alternative writings based on phonetic transcripts

References

R Harald Baayen, Richard Piepenbrock, and Leon Gulikers. The celex lexical database (release 2). *Distributed by the Linguistic Data Consortium, University of Pennsylvania*, 1995.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171--1179, 2015.

Gordon DA Brown and Richard PW Loosemore. A computational approach to dyslexic reading and spelling. In *Developmental and acquired dyslexia*, pages 195--219. Springer, 1995.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12): 3207--3220, 2010.

Max Coltheart. Lexical access in simple reading tasks. *Strategies of information processing*, pages 151--216, 1978.

Max Coltheart, Kathleen Rastle, Conrad Perry, Robyn Langdon, and Johannes Ziegler. Drc: a dual route cascaded model of visual word recognition and reading aloud. *Psychological review*, 108(1):204, 2001.

Timo Flesch, Jan Balaguer, Ronald Dekker, Hamed Nili, and Christopher Summerfield. Focused learning promotes continual task performance in humans. *bioRxiv*, page 247460, 2018.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128--135, 1999.

Karin Friedrich. Unterrichtskonzept und schriftspracherwerb. zum einfluss verschiedener pädagogisch-didaktischer konzepte auf lese-und rechtschreibleistungen, soziale kompetenzen und leistungsmotivation. *Pädagogische Hochschule. Zugriff am*, 22:2014, 2010.

Reinold Funke. Erstunterricht nach der methode lesen durch schreiben und ergebnisse schriftsprachlichen lernens--eine metaanalytische bestandsaufnahme. *Didaktik Deutsch*, 19 (36):20--41, 2014.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602--610, 2005.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369--376. ACM, 2006.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273--278. IEEE, 2013.

Yu Guan and Thomas Plötz. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):11, 2017.

Awni Hannun. Sequence modeling with ctc. *Distill*, 2(11):e8, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735--1780, 1997.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

Florian Hutzler, Johannes C Ziegler, Conrad Perry, Heinz Wimmer, and Marco Zorzi. Do current connectionist learning models account for reading development in different languages? *Cognition*, 91(3):273--296, 2004.

Karl-Heinz Jaeger. Vorlesung zur einführung in die sprachwissenschaft, 2010. see here.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, page 201611835, 2017.

Christian Klicpera, Alfred Schabmann, and Barbara Gasteiger-Klicpera. *Legasthenie-LRS*. UTB GmbH, 2013.

Dharshan Kumaran, Demis Hassabis, and James L McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7):512--534, 2016.

Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707--710, 1966.

Angelika Linke, Markus Nussbaumer, and Paul Portmann-Tselikas. Studienbuch linguistik. 5. erw. 2004.

Alena Lorenz. *Wer schreibt recht, wer schreibt schlecht?: Eine Untersuchung des Zusammenhangs zwischen der Rechtschreibleistung von Erstklässlern und dem Ansatz „Lesen durch Schreiben "von Jürgen Reichen in der Videostudie Deutsch des Projekts PERLE*, volume 27. kassel university press GmbH, 2017.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579--2605, 2008.

James L McClelland, Bruce L McNaughton, and Randall C O'reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

Christopher Olah. Understanding lstm networks. *GITHUB blog, posted on August*, 27:2015, 2015.

Charles E Osgood. The nature and measurement of meaning. *Psychological bulletin*, 49(3):197, 1952.

Joseph O'Neill, Barty Pleydell-Bouverie, David Dupret, and Jozsef Csicsvari. Play it again: reactivation of waking experience and memory. *Trends in neurosciences*, 33(5):220--229, 2010.

David C Plaut, James L McClelland, Mark S Seidenberg, and Karalyn Patterson. Understanding normal and impaired word reading: computational principles in quasi-regular domains. *Psychological review*, 103(1):56, 1996.

Jürgen Reichen. *Lesen durch schreiben*. sabe publisher, 1988.

Christa Röber-Siekmeyer and Helmut Spiekermann. Die ignorierung der linguistik in der theorie und praxis des schriftspracherwerbs. *Zeitschrift für Pädagogik*, 46(5):753--771, 2000.

Elke Sander. *Rechtschreibprobleme von Schülern am Ende der Grundschulzeit*. PhD thesis, Universität zu Köln, 2006.

Sascha Schroeder, Kay-Michael Würzner, Julian Heister, Alexander Geyken, and Reinhold Kliegl. childlex: A lexical database of german read by children. *Behavior research methods*, 47(4):1085--1094, 2015.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673--2681, 1997.

Mark S Seidenberg and James L McClelland. A distributed, developmental model of word recognition and naming. *Psychological review*, 96(4):523, 1989.

M Ali Basha Shaik. *Investigation on language modelling approaches for open vocabulary speech recognition*. PhD thesis, RWTH Aachen University, Germany, 2016.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104--3112, 2014.

Kyrana Tsapkini and Argye Hillis. Cognitive neuroscience of written language. In *The Oxford Handbook of Cognitive Neuroscience, Volume 1*. 2013.

Renate Valtin. Der" neue" methodenstreit oder:(was) können wir aus der amerikanischen leseforschung lernen. *Balhorn ua* (*1998, 63-80*), 1998.

Renate Valtin. Rechtschreiben lernen in den klassen 1-6. *Grundlagen und didaktische Hilfen*, 2000.

Margit Ergert Wien. Prosodie--schlüssel zur rechtschreibung. neue wege des schrifterwerbs1. *IDT 2013*, page 67, 2013.

Tal Yarkoni, David Balota, and Melvin Yap. Moving beyond coltheart's n: A new measure of orthographic similarity. *Psychonomic Bulletin & Review*, 15(5):971--979, 2008.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *arXiv preprint arXiv:1708.02709*, 2017.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Johannes C Ziegler, Conrad Perry, and Max Coltheart. The drc model of visual word recognition and reading aloud: An extension to german. *European Journal of Cognitive Psychology*, 12(3):413--430, 2000.