2020

KidSpell: A Child-Oriented, Rule-Based, Phonetic Spellchecker

Brody Downs
Boise State University

Oghenemaro Anuyah
Boise State University

Aprajita Shukla
Boise State University

Jerry Alan Fails
Boise State University

Maria Soledad Pera
Boise State University

See next page for additional authors

Authors

Brody Downs, Oghenemaro Anuyah, Aprajita Shukla, Jerry Alan Fails, Maria Soledad Pera, Katherine Wright, and Casey Kennington

Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6937–6946
Marseille, 11–16 May 2020

KidSpell: A Child-Oriented, Rule-Based, Phonetic Spellchecker

Brody Downs*, Oghenemaro Anuyah*, Aprajita Shukla*, Jerry Alan Fails*, Maria Soledad Pera*, Katherine Wright‡, Casey Kennington*
*Department of Computer Science & ‡Department of Literacy, Language and Culture
Boise State University, USA
{brodydowns, oghenemaroanuyah, aprajitashukla}@u.boisestate.edu,
{jerryfails, solepera, katherinewright, caseykennington}@boisestate.edu

Abstract

For help with their spelling errors, children often turn to spellcheckers integrated in software applications like word processors and search engines. However, existing spellcheckers are usually tuned to the needs of traditional users (i.e., adults) and generally prove unsatisfactory for children. Motivated by this issue, we introduce KidSpell, an English spellchecker oriented to the spelling needs of children. KidSpell applies (i) an encoding strategy for mapping both misspelled words and spelling suggestions to their phonetic keys and (ii) a selection process that prioritizes candidate spelling suggestions that closely align with the misspelled word based on their respective keys. To assess the effectiveness of KidSpell, we compare the model's performance against several popular, mainstream spellcheckers in a number of offline experiments using existing and novel datasets. The results of these experiments show that KidSpell outperforms existing spellcheckers, as it accurately prioritizes relevant spelling corrections when handling misspellings generated by children in both essay writing and online search tasks. As a byproduct of our study, we create two new datasets comprised of spelling errors generated by children from hand-written essays and web search inquiries, which we make available to the research community.

Keywords: spelling correction, children, phonetic, child spelling datasets

1. Introduction

Spelling is an essential literacy and life skill, the basics of which are taught to children during their first few years of school (Berkling, Kay, 2018). Cultivating spelling is considered to be a good reflection of children's understanding and learning of the alphabet, as well as a good predictor of their reading skills (Puranik et al., 2011). In teaching spelling skills to students, teachers traditionally center instruction based on children's five spelling development stages: precommunicative, semiphonetic, phonetic, transitional, and correct (Gentry, 1982). However, due to children's varying cognitive capabilities, some of them still struggle to spell as they progress in their education. To assist children in improving their learning abilities, teachers now incorporate the use of Assistive-Technology in the classroom (Simpson et al., 2009; Lawley, 2016), including spellcheckers that support spelling skill development. Spellcheckers are often embedded in software utilized in the classroom environment, e.g., word processors like Microsoft Word (Kutuzov and Kuzmenko, 2015), as well as search engines like Google (Wang and Zhao, 2019).

There are a number of spellcheckers, along with corpora, that can be leveraged in new developments (Kutuzov and Kuzmenko, 2015; Huang et al., 2013; Mitton, 2009). While these spellcheckers are tuned to the needs of traditional users, they prove unsatisfactory for children. Moreover, there is no empirical evidence on the applicability of these spellcheckers in handling children's misspellings–particularly during the time they are learning how to spell.

Children's spelling behaviors differ from adults irrespective of the spelling context–be it formulating queries (Gossen et al., 2011a) or writing essays (Greenberg et al., 2002). Compared to adults, children tend to use more phonological strategies (i.e., they usually spell using sounds) and less orthographic processes (i.e., memorizing letter sequences associated with individual words) (Greenberg et al., 2002). We therefore infer that spellchecking strategies introduced in Kutuzov and Kuzmenko (2015), Huang et al. (2013), and Mitton (2009) are inapplicable for children, as they focus on character or word sequences for mapping misspelled words to spelling corrections, without explicitly accounting for patterns unique to children. To address this, we introduce **KidSpell**, a rule-based, phonetic, English spellchecker tailored to the needs of children, aged 5–14 years.

The main goal of **KidSpell** is to provide relevant spelling suggestions that can capture children's spelling intent irrespective of the context of use. We show in three experimental settings, including hand-written short essays (Section 5.1.) and web search (Section 5.2.), that **KidSpell** outperforms existing, mainstream spellcheckers for correcting children's misspellings and is on-par with these tools when handling adult spelling errors (Section 5.3.).

The contributions of this work include:

- The design of a novel child-oriented English spellchecker[1] that is not only able to offer spelling suggestions that capture children's spelling intent, but also those that are suitable in the classroom,
- A corpus of spelling errors made by children, along with the correct spelling, made in hand-written essays and web search inquiries,
- A comparison of different spellcheckers, providing insights on limitations of existing tools when it comes to handling children's spelling errors, and
- An analysis that showcases the usefulness of **KidSpell** in handling children's spelling errors generated in both short essays or in web search environments.

2. Related Work

In the past decade, spellchecking strategies have been extensively studied. The work by Deorowicz and Ciura (2005)

[1] The source code for **KidSpell** and the described datasets can be found at https://github.com/BSU-CAST/KidSpell

outlines types of spelling errors as well as traditional strategies used to address those errors (e.g., edit distance, similarity keys, rule-based, probabilistic, and phonetic similarity). The authors in Mitton (2009) and Singh et al. (2016) also take advantage of edit distances as a ranking strategy, along with relative frequency for handing misspellings. Croft et al. (2015) designed a phonetic based strategy for mapping misspellings to phonetically similar spelling corrections. Similarly Mitton (2009) describes a model that first performs a simple dictionary lookup, then uses a non-phonetic key-generating algorithm, and finally considers homophone information to improve performance (e.g., *two* and *too*).

Machine learning approaches have also shown promise for designing spellcheckers. Kutuzov and Kuzmenko (2015) introduce a strategy that uses a morphological analyzer to find potential spelling errors in essays. Huang et al. (2013) explore the application of classifiers and neural networks, where edit distance and keyboard layout are the chosen features. Choe et al. (2019) use a neural grammar error correction system on a corpus of realistic errors from character sequences in words. De Amorim and Zampieri (2013) show unsupervised clustering of words as an alternative to edit distance. Whitelaw et al. (2009) present a similar approach, inferring knowledge about misspellings and word usage instead of labeled data.

It is important to consider spellcheckers in their context of use. Spellchecking strategies have been designed and evaluated for use in different contexts, such as hand-written essays (Mitton, 2009), keyboard spelling errors (Flor and Futagi, 2013; Kutuzov and Kuzmenko, 2015), short writing samples (Bassil, 2012), and queries written by users performing inquiries on search engines (Bassil, 2012; Sun et al., 2012; Li et al., 2011; Ganjisaffar et al., 2011). Because search tools are ubiquitously used and approximately 10-15% of queries formulated by online users consist of spelling errors (Gossen, 2015), it is a common domain for spelling errors resulting in implications for the results that are retrieved (e.g., some search engines fail to find results for misspelled queries (Fails et al., 2019)).

The web search domain lends itself to additional information that researchers have leveraged to improve spellchecking, such as query logs and sequence modeling using Ngrams in (Ganjisaffar et al., 2011) and (Li et al., 2011). The latter's proposed spellchecker – Cloud Speller – uses Hidden Markov Models leveraging Wikipedia data. Log information and sequential modeling are potentially useful for adult spelling error correction, but do not conform to child misspellings which are different from adult users, particularly in web search settings (Gossen et al., 2011b).

In the above-described papers, the focus is on adult spelling errors, whereas in our model we focus on child-specific spelling errors. To do so we build off the work of Croft et al. (2015) and Brill and Moore (2000), leveraging frequency-based and phonetic methods, as child spelling errors are not often captured with simple edit distance methods.

3. Model: KidSpell

In motivating the design of our model, we take inspiration from Deorowicz and Ciura (2005) who showed that the types of spelling errors that can be categorized as misspellings (e.g., pronunciation is known, but spelling is not) are commonly made by children and phonetic similarity approaches work well when correcting those mistakes. At a high level, given a misspelled term written by a child, our model applies a phonetic similarity approach in order to identify relevant spelling corrections that capture the child's information need. We illustrate the architecture of our spellchecking model in Figure 1.

3.1. Dictionary Creation

An important aspect of designing **KidSpell** involves creating a dictionary. For this purpose, we rely on the pre-compiled list of 100k most common words in the English language (Norvig, 2008), derived from the Google Web Trillion Word Corpus. From this list, we discard acronyms and non-English words, yielding approximately 40,000 unique words along with their frequencies (i.e., how common each word is in the English language). We augment our dictionary with words from the Age of Acquisition dataset (Kuperman et al., 2012), which is comprised of over 50,000 words, frequency, and the average age for which children first uttered those particular words. In total, our dictionary is comprised of 60,847 unique words.

3.2. Phonetic Spellchecking Approach

In this section, we discuss the process of encoding a phonetic key, using the key to gather candidates, and then ranking the top candidates to be used for spelling correction.

Phonetic Encoding. Following the method similar to the SoundEx model described by Croft et al. (2015), we use a phonetically-motivated approach to create a key that groups words that are similarly spelled or pronounced to use as spelling correction candidates. However, we implement a different phonetic encoding (as shown in the Appendix) to generate keys that are similar to the Metaphone algorithm (Philips, 1990), but is instead more focused on generating accurate phonetic representations, rather than grouping all potentially similar sounds. For instance, the letters *V* and *F* are not grouped together (i.e., implying that they make different sounds), nor are *Q* and *K*. Although they can make similar sounds, we found in our model development process that it was not common for children to use one of the letter pairs instead of the other. The encoding algorithm includes well-known phonetic rules such as recognizing that *ph* makes the *F* sound or the *k* in words starting with *kn* is silent. While vowels are used to determine the sounds of surrounding letters (e.g., *c* followed by *i*, *e*, or *y* makes the *S* sound), they are removed from the final key. This is due to their ambiguity as well as preventing tight groupings, resulting in several keys that only match a single word. Note that this encoding is focused on English spellings; adding new languages would require encoding rules for that particular language. To illustrate the phonetic encoding, consider the word *creature* mispelled as *crechur* (depicted in Figure 1), which **KidSpell** would process as follows:

- Recognize that the *t* in *ture* makes the *CH* sound (encoded as 1), transforming the word to *crea1ure*
- Recognize that as *c* is not followed by *i*, *e*, or *y*, it makes the *K* sound, resulting in *Krea1ure*

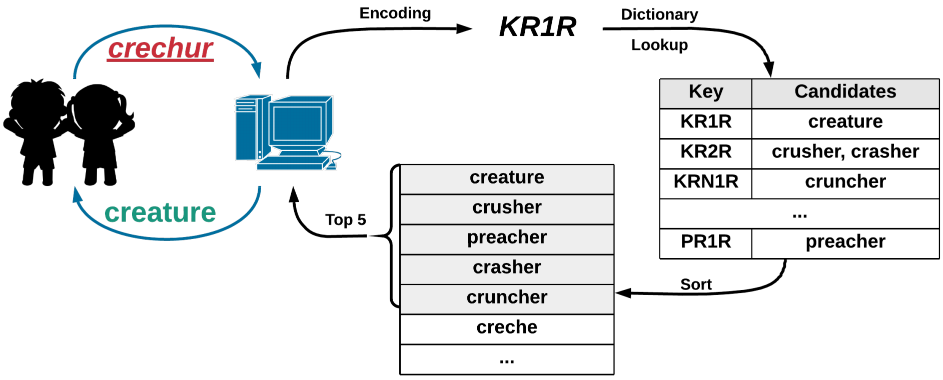

Figure 1: **KidSpell** architecture explained using *crechur* as an example misspelling for the word *creature*.

- Remove vowels, with the exception of *Y* at the end of a word, resulting in the final phonetic form: *KR1R*

As a pre-computational step, this process is performed on every word in our dictionary and a table is created that maps from the phonetic key to a list of words that match that key. For example, the key *NTRL* maps to a list containing the words *natural*, *neutral*, and *notarial*.

Selecting Candidates. Given a misspelled word, we encode a phonetic key and use that to identify words with similar keys. Using candidate suggestions that directly match the phonetic key of the misspelled word, we sort them based on their general frequency in the English language, similar to the approaches described in (Mitton, 2009) and (Singh et al., 2016). Recall that this frequency is a representation of how common the word is in the English language. More common words will be ranked higher.

Supplementary suggestions are found by looking up keys that are similar to that of the misspelled word, but not an exact match. For this, we generate keys that differ by an edit distance of 1, which involves removing a letter, adding a letter, substituting a letter for another, or transposing two adjacent letters. For instance, for the key *NTRL*, we also look up *NRL*, *NTRLD*, *NDRL*, and *NRTL*. Due to the larger number of supplementary suggestions, they are sorted by a normalized edit distance method rather than frequency. Candidates that exactly match the phonetic key of the misspelled word are prioritized over supplementary ones.

4. Spellcheckers and Datasets

In this section, we describe the spellcheckers and the datasets employed in our experiments.

4.1. Spellcheckers

In this study we examine a number of spellcheckers: *GingerIT*,[2] *Aspell*,[3] *Hunspell*,[4] *Bing*,[5] *SimpleSpellchecker (abbreviated SimSpell)*,[6] and *Enchant*.[7] We chose GingerIT with the aim of investigating how grammar-based spellcheckers perform when handling children's misspellings; Aspell and Hunspell due to their popularity as they have been adopted by a number of word processing tools like LibreOffice and OpenOffice, as well as web browsers. We chose Bing's spellchecker as it is designed specifically to work with a popular search engine preferred by children (Foss et al., 2012). Furthermore, SimSpell and Enchant spellcheckers' selection was prompted by the fact that they are popular open source spellchecking tools that can be used to complement existing software. We show that in Table 1, out of all the spelling suggestions provided by the above spellcheckers for the misspelled word *crechur*, none included the intended word *creature* on the list.

Table 1: Popular spellcheckers' spelling suggestions for the misspelled word *crechur*.

Spellchecker	Spelling suggestions
GingerIT	Crechur
Aspell	crotch, crusher, creches, Crecy, creche
Hunspell	Creche, church, Church
Bing	crechur
SimSpell	créche, creator, Creator, creamer, crasher
Enchant	creche, Church, church

4.2. Datasets

In our exploration, we use existing and newly-created datasets, which we have summarized in Table 2.

4.2.1. Children's Misspellings in Short Essays

We built a dataset based on writing samples from 49 children collected at a university-based literacy clinic. Each Fall and Spring semester, children in grades K–8 attend this clinic to receive one-on-one and small group tutoring from undergraduate students pursuing elementary education licensure. Children have diverse backgrounds, some are English language learners; some have learning disabilities.

All children independently complete a hand-written writing sample at the beginning of each semester. If a child's handwriting is difficult to decipher, tutors ask the child what they wrote and transcribe their writing, but no corrections are made to the child's original spelling. This process

[2] https://www.gingersoftware.com/spellcheck
[3] http://aspell.net/
[4] http://hunspell.sf.net
[5] https://tinyurl.com/AzureSpellcheck
[6] https://www.npmjs.com/package/simple-spellchecker
[7] https://pypi.org/project/pyenchant/

Table 2: Summary of misspelling datasets.

Dataset name	Audience	# of instances	Source	Attributes
Essay$_{\text{MSP}}$	Children	1,025	New	misspelled word, correct spelling, grade, spelling proficiency level
Wiki$_{\text{MSP}}$	Adults	2,455	Wikipedia Editor's dataset	misspelled word, correct spelling
Kids_Log$_{\text{MSP\_1}}$	Children	63	New	misspelled word, correct spelling, clicked word, suggestion, spellchecker, sessionID
Kids_Log$_{\text{MSP\_2}}$	Children	74	New	misspelled word, correct spelling, clicked word, suggestion, spellchecker, sessionID

is repeated throughout the semester to measure student progress. Each writing sample is transcribed digitally and annotated for potential spelling errors. We examined the writing samples collected over three years (i.e., six semesters) and recorded (i.e., digitized) each misspelled word as well as their intention. After removing duplicates, our resulting dataset consists of 1,025 misspelled words with their corresponding spelling corrections. We refer to this dataset as **Essay$_{\text{MSP}}$** [8] (see Table 3 for examples).

Table 3: Sample of instances in **Essay$_{\text{MSP}}$**.

Misspelled word	Correct spelling
favtit	favorite
somwan	someone
alectrek	electric

4.2.2. Children's Search Logs

Children are known to have more spelling errors than adults in web search queries (Gossen et al., 2011a), thus we also create datasets for this particular context. For this purpose, we performed two data collection studies with children during which we gathered misspelling data based on searches conducted in different environments. In both of these studies, children used a custom Search Interface (SI) on a desktop computer for submitting queries and were presented with the top-5 spelling corrections generated by a specific spellchecker (i.e., **KidSpell** or Aspell[9]). The SI relied on Google's API to power the retrieval of results in response to a child's query.

Children formulated queries based on verbal search prompts assigned by a facilitator (i.e., a graduate student or faculty). Their search tasks were formulated in order to enable children build upon information they had learned about Science, Technology, Engineering, and Mathematics (STEM) at school (see Table 4 for some sample search tasks and child-written queries). We assigned the same search task to each child in both studies, but allowed flexibility in scenarios where children decided to search for other classroom-related information. Facilitators took notes on how children interacted with the SI.

As shown in Figure 2, during query formulation, when a word is identified to be a spelling error, it is underlined and colored in red. Upon hovering over the misspelled word, a list of suggestions is provided from the respective spellchecker. Clicking on a suggestion replaces the misspelled word with the spelling suggestion. We record all these user and system actions. For both of these studies conducted, half of the computers used **KidSpell** and the other half, Aspell. Using these settings for our SI and spellchecking, we describe below the two studies where we collected data, which are based on different search environments.

Table 4: Example of search tasks along with a sample query written by children. Underlined words are misspelled.

Search tasks	Queries
Find an interesting fact about robots	wat is a robot made out of
Who was the first computer programmer?	the first compt
Who is the scientist that invented robots?	scientist who evented the robat

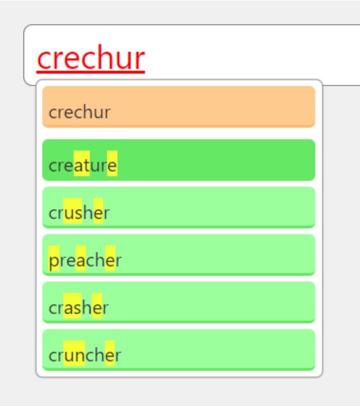

Figure 2: Example of **KidSpell** providing spelling suggestions in the custom SI.

Lab Search Environment. We performed the first data collection with children aged 6-11 years. The child participants are members of KidsTeam, an inter-generational design team that meets in a research lab twice a week (Druin, 1999; Guha et al., 2013; Fails et al., 2013). The goal of the team is for children and adults to work collaboratively as design partners to design technologies for children. Child participants vary from novice to intermediate in computer abilities. In this search environment, because the children had been meeting regularly they knew each other and were familiar with what they were expected to do as design partners, of which web search is included as part of the activities. Based on searches conducted, we created a dataset, **Kids_Log$_{\text{MSP\_1}}$**, with entries as a tuple per misspelled word containing (i) each word itself, (ii) the word that the child

[8] The dataset has additional information of grade and spelling proficiency levels.

[9] We selected Aspell as it obtains the best performance among traditional spellcheckers as demonstrated Section 5.1.

clicked on–one of the suggestions of the spellchecker that he/she thought of as relevant in each case, (iii) the correct spelling, which was collectively agreed upon by facilitators based on notes gathered, (iv) the associated session identifier (i.e., sessionID), (v) the spellchecker utilized, and (vi) the spelling suggestions offered.

Informal Search Environment. The second data collection was similar to the first in terms of search tasks assigned, facilitators, and the SI used. We focus on children aged 5–14 years, conducting searches at a local elementary school during a STEM event. From this, we created a dataset, **Kids_Log$_{MSP\_2}$**, that includes instances similar to **Kids_Log$_{MSP\_1}$**, but is instead based on queries gathered in this study. Note that for **Kids_Log$_{MSP\_2}$**, we excluded entries for which parents or peers intervened in the process of selecting spelling suggestions.

4.2.3. Adult Misspellings

We also use a dataset comprised of spelling errors made by adults. For this purpose, we take advantage of the Wikipedia misspelling corpus (denoted **Wiki$_{MSP}$**) which contains 2,455 misspellings of 1,922 words, along with the corresponding correct spellings. These misspellings were made by Wikipedia editors. Instances in **Wiki$_{MSP}$** are tuples of the form $<$ misspelled word, correct spelling $>$ (see Table 5 for some examples of instances in **Wiki$_{MSP}$**).

Table 5: Sample of instances in **Wiki$_{MSP}$**.

Misspelled word	Correct spelling
wendsay	wednesday
conquerer	conqueror
newyorker	new yorker

5. Experiments

In this section, we evaluate the performance of several spellcheckers in handling child-generated spelling errors in short essays (Section 5.1.) and search environments (Section 5.2.), as well as adult spelling errors (Section 5.3.).

5.1. Experiment 1: Child Spelling in Short Essays

In this experiment, we examine the effectiveness of **KidSpell** against other spellcheckers when handling misspellings generated by children while writing short essays.

Task. Given a misspelled word, each spellchecker is to (i) provide one or more suggestions for the correct spelling and (ii) rank those suggestions such that the highest ranked suggestion is the most suitable given a misspelled word.

Procedure. For each spelling error in **Essay$_{MSP}$**, we generate spelling suggestions from each spellchecker in our study and compare with respect to the gold standard (i.e., correct spelling in **Essay$_{MSP}$**) for performance analysis.

Metrics. To measure performance, we used Mean Reciprocal Rank (MRR) and Hit-rate. MRR captures the average position of the first relevant spelling suggestion in a list of ranked suggestions. The higher the MRR value is, the more effective the corresponding spellchecker is in terms of ranking the gold standard higher.

Hit-rate is used to determine the match between the gold standard and the suggestion list. In this case, the hit-rate value is 1 if a spellchecker has the gold standard in the suggestions list, otherwise, this value is 0. Taken together, these two metrics show how much the gold standard appears in the suggestion list (Hit-rate), and the rank of that word in the list (MRR). We report these metrics when varying the number of suggestions from 1-5 for the Hit-rate, and using the top-5 suggestions for the MRR.

Results. We summarize the results of this experiment in Table 6. **KidSpell** consistently outperforms other spellcheckers when it comes to finding the relevant spelling correction. Moreover, **KidSpell** is able to rank the relevant spelling correction approximately 74% of the time at the 2^{nd} position on average, making it easy for children to find them in a real scenario. Although Aspell had the best performance out of the traditional spellcheckers, both in terms of Hit-rate and MRR, it exhibits the limitations of these spellcheckers in handling children's misspellings, as on average, it is only able to find the relevant spelling for approximately 56% of the misspelled words and ranks these relevant spellings on the 3^{rd} position on average. The improvement of **KidSpell** over other spellcheckers, in terms of the MRR, is statistically significant (paired t-test, $p < 0.05$; n=1,025).

Table 6: Hit-rate in the top K (H@K) and MRR computed using **Essay$_{MSP}$**. K is the number of spelling suggestions provided by a spellchecker for each misspelled word.

K	**KidSpell**	Enchant	SimSpell	Bing	Hunspell	Gingerit	Aspell
				H@K			
1	**0.563**	0.217	0.180	0.305	0.210	0.370	0.247
2	**0.625**	0.283	0.265	0.330	0.286	0.370	0.348
3	**0.673**	0.360	0.302	0.336	0.350	0.370	0.461
4	**0.723**	0.412	0.330	0.336	0.400	0.370	0.523
5	**0.747**	0.462	0.364	0.336	0.445	0.370	0.569
				MRR			
	0.577	0.299	0.248	0.319	0.291	0.370	0.360

Investigating Inappropriate Words. It is imperative that spelling suggestions offered to this audience are appropriate for school-aged children. We therefore investigated the extent to which spelling suggestions generated by spellcheckers include sexually explicit or hate-based words. We determined words to be sexually explicit in nature if they exist in a dictionary of sexually explicit words created based on Google's bad words list.[10] Additionally, we identify hate-based words to be those that exist among the list of hate-speech and offensive language lexicons which we compiled from hateBase,[11] a repository of hate-speech language. We report in Table 7 the rate at which each spellchecker produced inappropriate words among the top-5 spelling suggestions.

We observed that **KidSpell** had a tendency to include sexually explicit and hate-based words among its top-5 spelling suggestions. Some sexually explicit or hate-based terms are ambiguous in nature and may not necessarily be inappropriate when considered in certain educational contexts. For

[10] https://code.google.com/archive/p/badwordslist/

[11] https://www.hatebase.org/

Table 7: Sexually explicit and hate-based word rate in top 5 suggestions. A light gradient indicates the best performance, while a darker shade implies worst.

KidSpell	Enchant	SimSpell	Bing	Hunspell	Gingerit	Aspell
Hate-based words						
0.0146	0.0156	0.0264	0.00293	0.0156	0.0039	0.0234
Sexually explicit words						
0.0538	0.045	0.0489	0.00097	0.04207	0.0078	0.0469

example, *slave* happens to be a derogatory term, but is also an important historical subject matter. In the classroom context, preventing children's exposure to false negative words (i.e., those that may not be inappropriate when considered as a single word, but tend to be inappropriate when surrounded by some context words) is less harmful than a false positive word (i.e., a word that may be relevant to the classroom but could potentially lead to the retrieval of inappropriate resources for children). Hence, for the rest of our analysis, we discarded from **KidSpell**'s dictionary, all the terms that exist in the sexually explicit and hate-based dictionaries.

5.2. Experiment 2: Child Spelling in Web Search Environments

In this section, we discuss two experiments conducted to compare **KidSpell**'s performance with that of Aspell (the best performing among traditional spellcheckers, as shown in Table 6). In the first experiment (Experiment 2A), we focus on the spellchecker's ability to handle spelling mistakes generated during search tasks conducted in a lab environment. For the second experiment (Experiment 2B), we focus on searches conducted in a classroom environment.

Task, Procedure, and Metrics. We used the task, procedure, and metrics introduced in Experiment 1, but on datasets **Kids_Log**$_{\text{MSP\_1}}$ and **Kids_Log**$_{\text{MSP\_2}}$. Note that in both experiments, the two spellcheckers flagged misspelled words that signalled to children that those words should be corrected, then generated respective spelling suggestions.

Results on Experiment 2A. As shown in Table 8, **KidSpell** outperforms Aspell both for offering the gold standard and in terms of ranking these high on the list of spelling suggestions. Per the results, Aspell was able to provide the gold standard at a rate of 56%, while ranking it at the 3^{rd} position on average. **KidSpell** surpasses Aspell on both metrics, providing the gold standard at a rate of 62% and ranking the word at the 2^{nd} position on average. However, the improvement in MRR of **KidSpell** over Aspell is not statistically significant (paired t-test; $p > 0.05$; n=74).

Table 8: H@K and MRR computed using **Kids_Log**$_{\text{MSP\_1}}$.

	MRR	H@K				
		1	2	3	4	5
KidSpell	**0.52**	**0.44**	**0.52**	**0.56**	**0.59**	**0.62**
Aspell	0.38	0.28	0.37	0.51	0.55	0.56

Results on Experiment 2B. As presented in Table 9, **KidSpell** outperforms Aspell both for offering the gold standards and ranking them high on the list of spelling suggestions. Similarly to Experiment 2A, children using Aspell would likely find the gold standard at the 3^{rd} position while **KidSpell**is able to rank the gold standards as the 2^{nd} spelling suggestion on average. The improvement of **KidSpell** over Aspell in terms of MRR is statistically significant (paired t-test, $p < 0.05$; n=63).

Table 9: H@K and MRR computed using **Kids_Log**$_{\text{MSP\_2}}$.

	MRR	H@K				
		1	2	3	4	5
KidSpell	**0.51**	**0.45**	**0.49**	**0.53**	**0.64**	**0.64**
Aspell	0.34	0.25	0.31	0.36	0.42	0.53

We further examine in both experiments if the spellchecker's ability to position the gold standards higher on the ranked list influences children's spelling suggestion selection and the propensity to find them. As shown in Table 10, most children selected spelling suggestions that were ranked first on the list. Upon further analysis, we found that children followed this selection pattern even when the gold standard was not on the suggestions list. Moreover, for cases where the gold standards were on the list of suggestions, children did not always click on them. For these particular datasets and tasks, we conjecture that the position of the word did not have any significant effect on a child's ability to find the correct word, though prior work has shown that ranking does influence choice (Gossen et al., 2011b).

Table 10: Exploration of the influence of ranking. *F* refers to the number of clicks at position K and *I* refers to how often the clicked word was the gold standard.

K	**Experiment 2A**		**Experiment 2B**	
	F	I	F	I
1	30	0.87	28	0.75
2	7	1.0	4	0.50
3	5	0.80	4	0.75
4	4	0.75	8	0.63
5	2	0.00	3	0.33

5.3. Experiment 3: Adult Spellings

In this experiment, we examine spellchecker performance when handling misspellings generated by adults.

Task, Procedure and Metrics. These are the same as the ones discussed for Section 5.1., though here we focus on adult misspellings using the **Wiki**$_{\text{MSP}}$ data.

Results. As shown in Table 11, among the spellcheckers, we observe that Enchant outperformed others both in terms of retrieving the gold standards and in assigning a high ranking for this spelling. We attribute Enchant's performance to the fact that it has the functionality to act as a comprehensive abstraction for dealing with different spellchecking libraries (e.g., both for Aspell and Hunspell) in a consistent way (Lachowicz, 2008) and its dictionary consists of words that target traditional audiences.

Table 11: H@K and MRR computed using **Wiki**$_{\text{MSP}}$.

K	**KidSpell**	Enchant	SimSpell	Bing	Hunspell	Gingerit	Aspell
H@K							
1	0.67	0.76	0.50	**0.81**	0.75	0.70	0.69
2	0.78	**0.87**	0.57	0.84	0.86	0.70	0.76
3	0.82	**0.90**	0.59	0.84	**0.90**	0.70	0.88
4	0.85	**0.91**	0.60	0.84	**0.91**	0.70	0.90
5	0.86	**0.92**	0.61	0.84	**0.92**	0.70	**0.92**
MRR							
	0.75	**0.83**	0.55	0.82	0.82	0.70	0.78

KidSpell was able to provide the correct spelling for more than 80% of misspelled words when the tool suggested at least three spellings. This indicates, that on average, adult users will be able to locate the right spelling correction for their misspelled words from among the top–3 suggestion offered by **KidSpell**. Surprisingly, even though GingerIT offers only a single spelling correction (reason for having the same results for both MRR and Hit-rate), it still outperformed SimSpell which obtained the worst performance among the spellcheckers. We conjecture that the poor performance of SimSpell is due to its strategy being reliant on a basic character-mapping approach (i.e., using metrics like the Levenshtein or edit-distance). The difference in MRR between **KidSpell** and all other spellcheckers is statistically significant (paired t-test; $p < 0.05$; n=2,455).

6. Discussion

Outcomes from the assessments presented in Section 5. show that **KidSpell** provides relevant spelling suggestions more than other spellcheckers for misspellings created by children, both in scenarios where they were formulating queries and writing short essays. When it comes to handling spelling errors generated by children in short essays, **KidSpell**'s ability to consistently prioritize gold standards indicates its applicability in text processing environments that are often used by this audience. Our results also show the effectiveness of **KidSpell** in prioritizing relevant spelling corrections for misspellings in children's search queries.

In a search scenario, being able to rank relevant spelling suggestions (i.e., those that capture the child's spelling intent) higher on the ranked list of spelling suggestions is imperative, as children are known to mostly select those spelling corrections that are ranked at the top of the list, even though they are not necessarily relevant (as demonstrated in the results presented in Section 5.2.). We attribute this outcome to the fact that children's reading and spelling skills are correlated (Plaza and Cohen, 2003), and as such, being that they are still developing spelling skills, they may experience difficulty determining the right spelling correction for their misspelled word. By using a query with the right terms (i.e., those that are spelled correctly) to initiate the search process on a search engine, this could translate to children being presented with resources that align with their search intent. This could also help to address issues with children's search task completion, as some search engines do not retrieve resources in response to misspelled queries (Fails et al., 2019). Hence, it is essential that child-oriented spellcheckers that can not only examine spelling patterns unique to children in order to capture spelling intent, but can also prioritize relevant spelling corrections on the list of offered spelling suggestions (i.e., a strength of **KidSpell**), are adopted by search engines utilized by this audience. Results also provide insights on the fact that **KidSpell**'s phonetic-based strategy is not only applicable to children's misspellings, as it is able to adequately map misspellings formulated by adults when the dictionary is robust.

In sum, outcomes suggest that relying on character-level strategies alone is not sufficient to align misspelled words to the their relevant spelling corrections for children. Instead, this can be better accomplished by examining aspects such as syntactical structure and phonetic patterns. Indeed, one important takeaway from this result is that children cannot rely on the general spellcheckers currently available and used. While state-of-the-art spellcheckers can successfully correct most of the adult spelling mistakes (i.e., as inferred from discussion in Section 5.3.), they fail to meet the needs of children, those who are in most need of spelling assistance. **KidSpell** is at-par with examined spellcheckers in correcting children's misspellings and while the model falls behind state-of-the-art spellcheckers on adult spelling mistakes, it is still competitive enough to be utilized as a general spellchecker that is effective for both adults and children.

7. Analysis

To scrutinize the kinds of errors **KidSpell** makes compared to other spellcheckers, we follow Deorowicz and Ciura (2005) and break down the spelling errors into two types: misspellings (pronunciation is known, but spelling is not) and mistypings (inserting or removing letters by accident). We then look at the efficiency of each spellchecker and finally, we examine the limitations facing our model.

Misspellings. The spelling errors in the short essay experiment described in Section 5.1. are defined by misspellings as these are all hand-written. The use of phonetic similarity techniques tend to work well when correcting misspellings (Deorowicz and Ciura, 2005) which explains our success in this experiment. Overall, our model most commonly succeeded where other spellcheckers failed when the spelling was at fault rather than the typing. Examples from the experiment in Section 5.1. include *brot* for *brought*, *rlrcsts* for *rollercoasters*, and *crecher* for *creature*. In these cases, **KidSpell** successfully retrieved and ranked the correct suggestion in the top 5 and the comparison spellcheckers did not. In terms of string manipulation, these words differ greatly from their gold standard, but can be considered phonetically similar.

Common among misspellings is the tendency to be unable to produce the correct grapheme for a vowel sound. Examples include *dun* for *done*, *thet* for *that*, and *grol* for *girl*. Again, in these cases **KidSpell** successfully retrieved the correct suggestion, while the comparison spellcheckers failed. Despite the difference of the misspelling and the gold standard being close in proximity, other spellcheckers often refused to give suggestions with differing vowels.

Mistypings. The use of a keyboard in the experiment described in Section 5.2. introduced mistypings. Spelling may be known, but the input medium made it more challenging to produce the intended spelling. In these cases, phonetic similarity is less likely to provide the correct suggestion as the mistypings did not match phonetically with the gold standard. For example, when given the mistyping *soudn*, our phonetic based model assumes the *d* sound should come before the *n*. **KidSpell** will find *sound* as a candidate but will not rank it as highly as other words such as *sudden*, *sadden*, and *sedan*. As such, our model generally did not perform as well as others on these types of errors.

We attribute the poorer performance on the adult dataset to the presence of mistypings, which frequently can be accounted for via a single edit (as noted by Deorowicz and

Ciura (2005)). The adult spelling errors made in the experiment described in Section 5.3., especially those made by Wikipedia editors, will less often be because of a lack of spelling knowledge, but because of an apparent mistype. **KidSpell** did not outperform its counterparts on words whose edit distance differed from the gold standard by one. However, it outperformed its counterparts on words of edit distance two or three.

When typing using a keyboard, we also see word boundary infractions (missing white spaces between two or more words as defined by (Lu et al., 2019). For instance, *tennisplayer* in place of *tennis player* and *hoomaderobots* in place of *who made robots*. These types of errors were rare in the experiment described in Section 5.1. (where hand-written writing was used) as the spacing between words and letters may not be consistent and may be assumed when that wasn't the intention. Fixing spelling errors via word splitting is a task some spellcheckers perform but is not a feature included in our model.

Efficiency. For deployment and scaling purposes, it is important to examine the speed at which a spellchecker can correct misspellings. In Table 12 we report the rate at which each of the spellcheckers can find 5 suggestions for the given words, each in the same environment. These values were determined by evaluating the 1,025 words from the experiment in Section 5.1. **KidSpell** performs slightly above the other spellcheckers, creating suggestions for words at a rate of 22.6 words per second.

Table 12: Rate of words processed per second using **Essay$_{\text{MSP}}$**. A light gradient indicates the best performance, while a darker shade implies worst.

KidSpell	Enchant	SimSpell	Bing	Hunspell	GingerIT	Aspell
22.6	15.6	19.9	1.2	20.1	1.4	21.9

Other Limitations. Some spelling errors could not be corrected due to the gold standard not appearing in the model's dictionary. This was the case for 5% of the gold standards when correcting the adult spelling errors as that data set involved especially uncommon words or proper nouns. For example, **KidSpell** could not retrieve *Athenians* or *Bernoulli* as suggestions due to neither of these words being found in our dictionary. We also see pop culture terms in our experiments with child users (e.g., *Optimus Prime* and *BattleBots*). All the spellcheckers failed to produce these words as they were not part of their respective dictionaries. However, adding these words to the dictionary would be a trivial modification.

An error introduced in the search environment is the attempt to use the spellchecker as an auto-complete feature (e.g., *ro* for *robot* then *robi* for robot). Neither **KidSpell** nor Aspell handled these types of errors successfully.

In the search environment, the spellcheckers failed to detect real word errors (i.e. errors that result in the correct spelling of an unintended word - e.g., *wit* for *what*). This was a result of the method used for spelling error detection, which was to look up each word in a dictionary. These types of errors are often due to incorrect vowel usage, which the phonetic method used in **KidSpell** tended to work well with.

Spelling errors where none of the spellcheckers could pick up the intention featured particularly bad spelling. In some cases, it seems they may know most of the letters in the word, but didn't know the ordering (e.g., *tetcnoglye* for *technology*). In other cases, there was a lack of understanding the pronunciation or an issue with mapping phonemes to graphemes (e.g., *peroger* for *programmer*).

A shortcoming for **KidSpell** is in its ranking of candidates generated for shorter words, which often have phonetic keys that are common with many others. For example, our model failed to suggest *flute* for the misspelling *flut* despite their matching keys. In this case, 5 words were suggested ahead of flute: *flat*, *felt*, *fight*, *fault*, and *float*. In the short essay experiment described in Section 5.1., there were 54 cases where our model failed, but Aspell succeeded in suggesting the gold standard. In 50 of those cases, our model retrieved the candidate successfully, yet failed to rank it within the top 5 suggestions. For all spelling errors in that experiment, the correct candidate is found for 90% of the misspelled words, but is only ranked among the top 5 for 74% of them.

8. Conclusions and Future Work

We introduced **KidSpell** a phonetic, rule-based spellchecker that corrects spelling errors generated by children. Experiments based on essay and online search environments demonstrated that **KidSpell** outperforms well-known, general-purpose counterparts considered for analysis when applied to detect and correct child misspellings. Our experiments also showcase that the performance of the proposed model is comparable to existing counterparts when correcting adult spelling mistakes, enabling the use of **KidSpell** as a general spellchecker for a diverse audience. Part of our contribution are new, freely available datasets of child spelling errors. To the best of our knowledge, they are the first of their kind and comprise of spelling errors in various environments from children in different contexts - hand-written essays (grade K-8) and web search environments (age 5-14 years) .

In future work, we will enhance our ranking algorithms, possibly by using features based on age of acquisition (discussed in Section 3.1.) and word frequency. As **KidSpell** only performs single word evaluations, techniques that consider the context of the given words could further improve our ability to correctly rank candidates. In order to improve the ability of the model in environments where a keyboard is used, we could use keyboard layout information and consider mistypings more strongly over misspellings.

With the knowledge that our phonetic based model works better with children's spelling mistakes, other known models could be explored in combination with phonetic approaches which would have implications for children's search engine designers, educators, and researchers in the field of NLP.

Acknowledgements

We would like to thank the children who are part of *kidsTeam* for their cooperation and participation in the data collection study. We also thank the organisers of the STEM event at a local school, for inviting us to show our system and collect data for evaluation. This research was primarily funded by the National Science Foundation (Award # 1763649).

9. References

Bassil, Y. (2012). Parallel spell-checking algorithm based on yahoo! n-grams dataset. *arXiv preprint arXiv:1204.0184*.

Berkling, Kay. (2018). A 2nd longitudinal corpus for children's writing with enhanced output for specific spelling patterns. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

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

Choe, Y. J., Ham, J., Park, K., and Yoon, Y. (2019). A neural grammatical error correction system built on better pre-training and sequential transfer learning. *arXiv preprint arXiv:1907.01256*.

Croft, W. B., Metzler, D. B., and Strohman, T. B. (2015). *Search Engines: Information Retrieval in Practice*. Pearson Education, Inc.

De Amorim, R. C. and Zampieri, M. (2013). Effective spell checking methods using clustering algorithms. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 172–178.

Deorowicz, S. and Ciura, M. (2005). Correcting spelling errors by modelling their causes.

Druin, A. (1999). Cooperative Inquiry: New Technologies for Children. page 8.

Fails, J. A., Guha, M. L., Druin, A., et al. (2013). Methods and techniques for involving children in the design of new technology for children. *Foundations and Trends® in Human–Computer Interaction*, 6(2):85–166.

Fails, J. A., Pera, M. S., Anuyah, O., Kennington, C., Wright, K. L., and Bigirimana, W. (2019). Query formulation assistance for kids: What is available, when to help & what kids want. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, IDC '19, pages 109–120, New York, NY, USA. ACM.

Flor, M. and Futagi, Y. (2013). Producing an annotated corpus with automatic spelling correction. *Twenty Years of Learner Corpus Research: Looking back, Moving ahead. Corpora and Language in Use, eds. S. Granger, G. Gilquin and F. Meunier*, pages 139–154.

Foss, E., Druin, A., Brewer, R., Lo, P., Sanchez, L., Golub, E., and Hutchinson, H. (2012). Children's Search Roles at Home: Implications for Designers, Researchers, Educators, and Parents. *Journal of the Association for Information Science and Technology*, 63(3):558–573.

Ganjisaffar, Y., Zilio, A., Javanmardi, S., Cetindil, I., Sikka, M., Katumalla, S., Khatib, N., Li, C., and Lopes, C. (2011). qspell: Spelling correction of web search queries using ranking models and iterative correction. In *Spelling Alteration for Web Search Workshop*, page 15.

Gentry, J. R. (1982). An analysis of developmental spelling in gnys at wrk. *Reading Teacher*.

Gossen, T., Low, T., and Nürnberger, A. (2011a). What are the real differences of children's and adults' web search. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1115–1116, New York, NY, USA. ACM.

Gossen, T., Low, T., and Nürnberger, A. (2011b). What are the real differences of children's and adults' web search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1115–1116. ACM.

Gossen, T. (2015). Large-scale analysis of children's queries and search interactions. In *Search Engines for Children*, pages 79–85. Springer.

Greenberg, D., Ehri, L. C., and Perin, D. (2002). Do adult literacy students make the same word-reading and spelling errors as children matched for word-reading age? *Scientific Studies of Reading*, 6(3):221–243.

Guha, M. L., Druin, A., and Fails, J. A. (2013). Cooperative Inquiry revisited: Reflections of the past and guidelines for the future of intergenerational co-design. *International Journal of Child-Computer Interaction*, 1(1):14–23, January.

Huang, Y., Murphey, Y. L., and Ge, Y. (2013). Automotive diagnosis typo correction using domain knowledge and machine learning. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 267–274. IEEE.

Kuperman, V., Stadthagen-Gonzalez, H., and Brysbaert, M. (2012). Age-of-acquisition ratings for 30,000 english words. *Behavior research methods*, 44(4):978–990.

Kutuzov, A. and Kuzmenko, E. (2015). Semi-automated typical error annotation for learner english essays: integrating frameworks. In *Proceedings of the 4th workshop on NLP for Computer Assisted Language Learning at NODALIDA 2015, Vilnius, 11th May, 2015*, number 114, pages 35–41. Linköping University Electronic Press.

Lachowicz, D. (2008). Enchant.

Lawley, J. (2016). Spelling: computerised feedback for self-correction. *Computer Assisted Language Learning*, 29(5):868–880.

Li, Y., Duan, H., and Zhai, C. (2011). Cloudspeller: Spelling correction for search queries by using a unified hidden markov model with web-scale resources. In *Spelling Alteration for Web Search Workshop*, pages 10–14. Citeseer.

Lu, C. J., Aronson, A. R., Shooshan, S. E., and Demner-Fushman, D. (2019). Spell checker for consumer language (CSpell). *Journal of the American Medical Informatics Association*, 26(3):211–218, 01.

Mitton, R. (2009). Ordering the suggestions of a spellchecker without using context. *Natural Language Engineering*, 15(2):173–192.

Norvig, P. (2008). Natural language corpus data: Beautiful data.

Philips, L. (1990). Hanging on the metaphone. *Computer Language Magazine*, 7(12):39–44, December.

Plaza, M. and Cohen, H. (2003). The interaction between phonological processing, syntactic awareness, and naming speed in the reading and spelling performance of first-grade children. *Brain and cognition*, 53(2):287–292.

Puranik, C. S., Lonigan, C. J., and Kim, Y.-S. (2011). Contributions of emergent literacy skills to name writing, letter writing, and spelling in preschool children. *Early childhood research quarterly*, 26(4):465–474.

Simpson, C. G., McBride, R., Spencer, V. G., Lodermilk, J., and Lynch, S. (2009). Assistive technology: Supporting learners in inclusive classrooms. *Kappa Delta Pi Record*, 45(4):172–175.

Singh, S. P., Kumar, A., Singh, L., Bhargava, M., Goyal, K., and Sharma, B. (2016). Frequency based spell checking and rule based grammar checking. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 4435–4439. IEEE.

Sun, X., Shrivastava, A., and Li, P. (2012). Fast multi-task learning for query spelling correction. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 285–294. ACM.

Wang, C. and Zhao, R. (2019). Multi-candidate ranking algorithm based spell correction. In *The 2019 SIGIR Workshop On eCommerce*, Paris, France.

Whitelaw, C., Hutchinson, B., Chung, G. Y., and Ellis, G. (2009). Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 890–899. Association for Computational Linguistics.

Appendix

The following table describes the ruleset used to transform a word into a phonetic key. The symbols used are similar to those in regular expressions: ^ indicates the beginning of a word, $ indicates the end of a word or common suffixes, * indicates 0 or more for the previous letter, ! is a Boolean NOT. Letters inside parenthesis are not changed while those inside square brackets indicate a match for any letters in the set. A result of _ indicates deletion. *th* is encoded as 0, *ch* is encoded as 1, *sh* is encoded as 2.

Table 13: Ruleset used to transform a word into a phonetic key.

Phonetic ruleset			
Substring	**Result**	**Substring**	**Result**
ck	K	[st](i[ao])	2
^[aeiou]	A	s*c(iey)	S
^[gpk]n	N	c	K
^wr	R	dg([iey])	J
^gh	G	gh(![aeiou])	_
(^s)ch	K	gh$	_
mb$	M	gh	G
^y	Y	gn$	N
th	0	y$	Y
t*ch	1	ph	F
t(ure)	1	[hwy](![aeiou])	_
sh	2	z	S
c(ion|iou)	2	[aeiou]	_