International Journal of Child-Computer Interaction 32 (2022) 100373

Contents lists available at ScienceDirect

International Journal of Child-Computer Interaction

journal homepage: www.elsevier.com/locate/ijcci

Research paper

KidSpell: Making a difference in spellchecking for children

Brody Downs [a,*], Maria Soledad Pera [a], Katherine Landau Wright [b], Casey Kennington [a], Jerry Alan Fails [a]

[a] Department of Computer Science, Boise State University, USA
[b] Department of Literacy, Language and Culture, Boise State University, USA

ARTICLE INFO

Article history:
Received 2 December 2020
Received in revised form 21 July 2021
Accepted 2 August 2021
Available online 9 August 2021

Keywords:
Spelling correction
Spellchecking
Children
Search engines
Spelling cues

ABSTRACT

Children's ability to spell effectively is a major barrier to using search engines successfully. While search engines make use of spellcheckers to provide spelling corrections to their users, they are designed for more traditional users (i.e., adults) and have proven inadequate for children. The specific target of children for this research are those with early literacy skills (whose are typically ages 6–12). The aim of this work is twofold: first, to address the types of spelling errors children make by researching, developing, and evaluating algorithms to generate and rank candidate English spelling suggestions for children; and second, to improve children's user experience when using our proposed spellchecker by involving them in the design process through participatory design and evaluating the impact of interactive elements on children's spellchecking behaviors. The outcomes of our studies and assessments result in a phonetic-based spelling correction model (KidSpell) that can more accurately correction children's spelling errors than existing state-of-the-art models. Further, we learned that visual and audio cues have a positive impact on children's ability to find their intended word from a list of spelling suggestions.

1. Introduction

Children's use of search tools, including popular search engines like Bing or Google, for information discovery is a common task (Azpiazu, Dragovic, Pera, & Fails, 2017). When typing the queries that are meant to initiate these searches, it is often the case that spelling errors occur. Without adequate spelling correction, the presence of spelling errors in a query can cause search engines – both commercial and those dedicated to children – to not only retrieve resources irrelevant to the user's information needs, but may also result in empty search engine result pages (this latter is more prominent on children's search engines) (Fails et al., 2019; Wang & Zhao, 2019). This is especially problematic for children considering reports indicating that between 25% and 40% of queries formulated by children, ages 6 to 13, contain at least one spelling error (Gossen, Low, & Nürnberger, 2011). While there is extensive research focusing on correcting the spelling of queries, spellcheckers are usually based on past query logs leading to spelling suggestions that often better resonate with a general (i.e., adult) audience because that data is more readily available (Downs et al., 2019). Furthermore, the spelling strategies that children use differ from adults as they tend to use phonetic strategies (i.e., using sounds) rather than orthographic ones (i.e., memorizing letter sequences) (Greenberg, Ehri, & Perin, 2002). Even when children are taught language with different pedagogical approaches (e.g. phonics or not), children are more likely to use invented spelling (which is encouraged) (Gentry, 2000). Supporting children's misspelling corrections requires a model built from the ground up that generates suitable spelling candidates and ranks them appropriately.

Even if child spelling errors could be more accurately corrected, the design of spelling interfaces that children find intuitive is a non-trivial problem. Modern interfaces often aim to correct spelling errors quickly, efficiently, and automatically, rather than helping users develop spelling skills, which conflicts with research-based best practices for spelling instruction (Joshi, Treiman, Carreker, & Moats, 2008). In an interactive system, spelling suggestions are presented to the user from which they then choose the suggestion that correctly matches their intended word. However, when children are presented with spelling suggestions, the word they click does not always match the word they intended to type (Downs, Anuyah et al., 2020) because they simply do not know the correct spelling of the intended word. These findings are likely due to spelling and reading development being highly correlated (Bear, Invernizzi, Johnston, & Templeton, 1996), so a word that is difficult for a child to spell will likely be difficult for them to read and identify from a list of similar words. In similar tasks, children have also shown a propensity to interact with higher-ranked alternatives (Anuyah, Fails, & Pera, 2018;

* Corresponding author.
E-mail address: brodydowns@u.boisestate.edu (B. Downs).

https://doi.org/10.1016/j.ijcci.2021.100373
2212-8689/

Gwizdka & Bilal, 2017), meaning they often choose from the higher listed alternatives even when those options do not match their intent. Although there is little known research regarding children's interactions with spellchecking interfaces, the use of multimodal cues (e.g., audio and visual) have been shown to be preferred by children and aid in reading comprehension when compared to using just a single modality (e.g., only text) (Druin, Foss, Hutchinson, Golub, & Hatley, 2010; Gossen, Nitsche, & Nürnberger, 2012; Sadoski, Goetz, & Fritz, 1993).

In this study, we discuss research advances made to address the issues children face with English spelling correction when interacting with a spellchecking interface, especially in web search settings. Our research work is informed by and responds to children ages 6–12 with varying spelling developmental levels. Additionally while the focus is on English, we also consider children with different language backgrounds (for more information about these considerations see Sections 3.1 and 5.4). The work presented in this study seeksto answer the following questions:

- Do spelling correction algorithms that align with children's spelling behaviors (e.g. phonetic) improve spelling candidate generation? (Section 4.1)
- What machine learning models work best for re-ranking spelling suggestion candidates? (Section 4.2)
- Which input features are most effective for ranking spelling suggestion candidates for children? (Section 5.2)
- How does KidSpellλ (the spellchecker for children described in this study) compare to other baseline spellcheckers with regards to children's grade (with approximate ages), spelling development level, and native language? (Section 5.4)
- Are audio and visual cues effective in assisting children in making spelling suggestion selections in an interactive spellchecker? (Section 6.1)
- What design features to children prefer in a spellchecking interface? (Section 6.2)

To answer these questions, we present a candidate generation model, Kidspell, and improve that with a complete spelling correction model, KidSpellλ, which leverages known children's spelling habits and data pertaining to children's spelling errors to produce more suitable corrections for children. We then analyze and evaluate the use of multimodal cues as well as further explore other problems and solutions to spellchecking through participatory design. Motivated by children's phonological strategies to spell, we use a phonetic encoding strategy to map words and misspellings to phonetic keys to effectively and efficiently provide spelling correction candidates. We then research the relative advantages of different machine learning models, with a focus on Learning To Rank (LTR) and features designed towards a child user to improve ranking. These methods are extensively evaluated against state-of-the-art models both for generating and ranking candidates. Experimental results show KidSpellλ is able to more accurately provide and rank spelling corrections when handling misspellings generated by children in both essay writing and web search settings. In addition, we demonstrate improved spelling correction across different ages, spelling development level, and native languages of the users when compared to state of the art models. We further design and evaluate the impact of visual and audio cues on children's selection habits and show a positive impact on assisting children in selecting correct spelling suggestions (see Fig. 1 for an example of KidSpell with picture cues). Through the use of participatory design involving children as design partners, we discover other issues related to spellchecking interfaces and propose steps going forward.

The main contributions of this work are the design of a novel child-oriented spelling correction tool and the study of the effectiveness of media cues in a spellchecking interface for children (ages 6–12). Analysis and evaluations provide insights on the limitations of existing tools when it comes to handling children's spelling suggestions. Outcomes from this work have potential broader impacts to search and spellcheckers for children. The spelling correction data and algorithms used are made publicly available.[1]

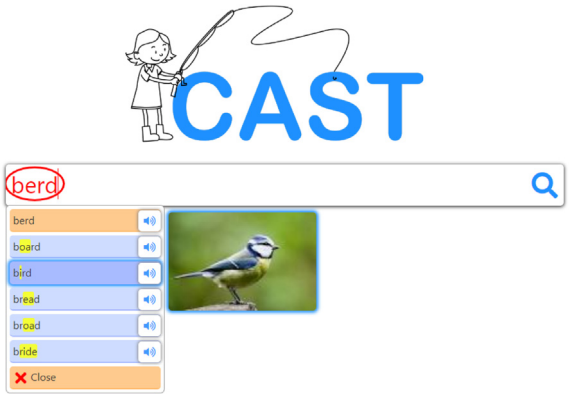


Fig. 1. Example of KidSpell being used in the CAST (Child Adapted Search Tool) interface, designed for children ages 6–12.

In the remainder of this study, we first discuss related work (Section 2) in relation to spelling behaviors, spelling correction methods, and creating interfaces for children. This is followed by a discussion on data collected on children's spelling errors necessary for our studies (Section 3). We then describe our spelling correction method which involves phonetic candidate generation and machine learning based ranking (Section 4). An in-depth evaluation and analysis is then completed on the correction method (Section 5). Thereafter, we present studies examining the impact of multimodal cues (images and audio) on children's selection of the intended misspelled word with a discussion of these results (Section 6.1). We then present participatory design sessions that built on lessons learned from this study and sought to further enhance the spellchecker user interface (Section 6.2). Finally, we offer concluding remarks and directions for future work (Section 7).

2. Related work

In this section, we review related work focused on English language spelling strategies of children, in addition to spelling errors made by children and how they compare to adults. Thereafter, we discuss state-of-the-art methods for correcting spelling errors. Lastly, we summarize prior work addressing children's interactions with computers and how that can be used to guide our design for a more effective spellchecking tool.

2.1. Spelling strategies and errors

To approach spelling correction it is necessary to understand the spelling process and the types of spelling errors being made, which then allows us to take steps to undo those errors (Deorowicz & Ciura, 2005). Greenberg et al. (2002) reported that when compared to adults, children (grades 3–5 in the United States, approximate ages 8–11) tend to use more phonological strategies (i.e., spelling by using sounds) and fewer orthographic processes (i.e., memorizing letter sequences associated with individual words). As a result, the spelling errors made by children

[1] https://github.com/BSU-CAST/KidSpell.

often differ from adults, which spellcheckers often do not consider. Deorowicz and Ciura (2005) reported on three different types of spelling errors that occur in the spelling and typing process: vocabulary incompetence (e.g., *unperfect* instead of *imperfect*), misspellings (e.g., *grammer* instead of *grammar*), and mistypings (e.g., *spwlling* instead of *spelling*). Typing errors for children (ages 7–11) have been noted, particularly that typing can cause children to end the search task prematurely or lead to not being able to find the results they intended (Druin et al., 2010). Although it has been noted that misspellings and vocabulary incompetence are common types of errors formed by children (Deorowicz & Ciura, 2005), no studies have shown how correction tools that focus on these types of errors are particularly useful for children. This inspires us leverage in the design of KidSpell algorithms that are more focused on addressing these errors.

2.2. Spelling correction

Little research has been done on spelling correction that targets a child audience or the effectiveness of spelling correction strategies on children's spelling errors. In an overview of traditional methods, Deorowicz and Ciura (2005) note that phonetic similarity methods are effective at correcting *misspellings*, an error type commonly made by children. Phonetic similarity strategies, such as SoundEx (Croft, Metzler, & Strohman, 2010) and PHONIX (Gadd, 1990), use similarity keys techniques to produce words with similar phonetic pronunciation in an effort to find similar surnames. Other algorithms to produce phonetic keys to index words, such as Metaphone (Philips, 1990), have been used in the Aspell spellchecker (Aspell, 2020), a common and popular baseline in spelling correction research. While effective at correcting *misspellings*, none of the aforementioned works are tuned for children or for general-purpose spellchecking (Deorowicz & Ciura, 2005).

More recently, machine learning methods have been applied to correct spelling errors. The work by Pande (2017) and De Amorim and Zampieri (2013) leverage character string embeddings and unsupervised clustering for candidate generation to quickly generate candidates for correction. Other machine learning methods have focused on ranking or candidate selection. Fomin and Bondarenko (2018) and Huang, Murphey, and Ge (2013) make use of classifiers and comprehensive feature sets. Researchers instead seeking to improve spelling correction in search engines have used language model such as hidden Markov models (Li, Duan, & Zhai, 2011) and various n-gram language models (Ganjisaffar et al., 2011). Common in each of these machine learning methods is that the focus is on adult users and the data used is consists of adult spelling errors. Many make use of features that are effective for adult spelling errors, such as simple edit distance metrics, which are ineffective when correcting children's spelling errors (ages 6–11) (Downs, Anuyah et al., 2020). As a result, they do not conform to the types of errors children make, particularly in web search settings (Gossen et al., 2011) (children's ages not identified). Furthermore, the training of many of these state-of-the-art methods require a corpus of spelling error data which is largely unavailable for children.

2.3. Children's interaction

While it is an important first step to be able to accurately correct spelling errors made by children, engagement is key to improve the experience children have when interacting with a spellchecker. A study by Figueredo (2006) explored the use of a spellchecker by children (grades 4 and 6) for story writing composition rather than search. They found that children frequently used spellcheckers to correct spelling and were mostly successful when correcting errors using a spellchecker. An investigation by Druin et al. (2010) into children's interactions (children ages 7–11) with search engines documented children's trouble with spelling and typing and advocated for interactive spelling assistance in web search. Many other studies on children's interactions when searching online also report on their difficulty with spelling (Fails et al., 2019; Gossen et al., 2011, 2012; Landoni, Matteri, Murgia, Huibers, & Pera, 2019). While little research has been done exploring children's experience with interactive spellcheckers, we look to the work done in the field of child–computer interaction to assist in the design.

The use of participatory design techniques, where children are involved in the design process, allow children to have a voice in design of new technologies (Fails et al., 2013; Guha, Druin, & Fails, 2013; Hourcade, 2015; Nesset & Large, 2004). We make use of participatory design techniques to identify problems, generate solutions, and further understand children's needs when it comes to spellchecking.

Hourcade (2015) emphasizes the importance of speaking the user's language when designing interfaces, which for children may not be text, but sounds and images. Researchers have shown children's preference for visual interfaces (ages 11–14) (Kuhn, Cahill, Quintana, & Schmoll, 2011) and multiple types of input (e.g., images) (children ages 7–11) (Druin et al., 2010; Gossen et al., 2012). Additionally, Sluis et al. (2004) used sounds/phonemes to enforce reading skills and Michaelis and Mutlu (2019) used synthesized speech with textbooks to boost children's interest and understanding (children ages 10–12). These findings also align with Dual Coding Theory (Sadoski et al., 1993) (for adults) which posits that providing information in multiple modalities aids readers' comprehension. While children's preferences and attentiveness towards visual or audio cues have been documented (Hourcade, 2015; Kuhn et al., 2011; Sluis et al., 2004), there has yet to be research that employs these methods in the context of a spellchecking interface.

2.4. Ranking

The order in which suggestions are displayed is another important aspect impacting the children's selection. The ranking of vertically positioned options influences children's choice (Duarte Torres, Hiemstra, & Huibers, 2013; Gossen et al., 2011), showing a bias of options that are oriented higher. Similarly, Downs, Anuyah et al. (2020) demonstrated children tend to favor the higher-ranked spelling suggestions, regardless if they are correct. This emphasizes the need for not only finding correct spelling suggestions but ranking them appropriately.

3. Method: Data collection

Essential to the models and evaluations presented in this study is a collection of child-made spelling errors. An important reason for this is that misspellings are complex: sometimes stemming from typographical errors ('teh' for 'the'), but oftentimes the reasons are more complex and related to cognitive misunderstandings of the spelling of words (Bear et al., 1996; Chen, Li, & Zhou, 2007; Joshi et al., 2008). Due to this complexity we use spelling errors collected from children in two different contexts: hand-written essays and typed search queries, which we have summarized in Table 1.

Table 1
Summary of children's spelling error datasets. The attributes identified include the data that was available from each of the respective datasets.

Dataset name	# of misspellings	Source	Attributes
$\text{Essay}_{\text{MSP}}$	1651	Hand-written essays	misspelled word, correct spelling, grade, spelling level, native language, words before
$\text{Query}_{\text{MSP}}$	134	Typed search queries	misspelled word, words before misspelled word, correct spelling, selected suggestion, sessionID
$\text{Childrens}_{\text{MSP}}$	1785	Combination of $\text{Essay}_{\text{MSP}}$ and $\text{Query}_{\text{MSP}}$	misspelled word, correct spelling, words before misspelled word

Table 2
Sample subset of instances in $\textbf{Essay}_{\text{MSP}}$. The dataset only included the child's grade (not their age). The typical ages of children in the United States for Grade 3 is 8–9; and for Grade 4, 9–10.

Target	Spelling	Grade	Spelling development level	Language	Words before
always	olwes	4	Letter Name Alphabetic	Spanish	like she
differences	diffrences	3	Syllables and Affixes	Korean	similarities and
professor	pfes	3	Within Word Pattern	English	was it

3.1. Children's spelling errors in hand-written essays

We built a hand-written essay spelling error dataset based on writing samples from 82 children (grades K-8 in the United States; approximately ages 5–14) with diverse backgrounds. Although some children are outside of our target age range (6–12) we evaluate our spelling correction on their data as well. The majority were identified by their parents as struggling with literacy development in English and many were English language learners. The distribution of native languages of children who provided writing samples included: 42 English, 30 Korean, 2 Italian, 2 Spanish, 2 Japanese, and 1 Mandarin.

This group of children were all participating in an after school, University-based literacy tutoring program. This ongoing program is offered each fall and spring semester, and provides an opportunity for undergraduate students in teacher preparation programs to gain tutoring experience under the supervision of University faculty. Children are enrolled each semester on a first come, first serve basis, and the enrolled population generally reflects the demographics of the surrounding community. All of the children in this study were attending school where English was the language of instruction. Most classrooms where English is the language of instruction contain diverse populations of students with varied language backgrounds (Uro & Lai, 2019). In order to ensure that Kidspell would be effective for the vast majority of learners, and thus allow us to make generalizations about our findings, it was necessary to include children from different language backgrounds.

Writing samples were collected at a university-based literacy clinic where children receive one-on-one and small group tutoring from undergraduate students pursuing elementary education licensure. At the beginning of each semester, tutors administer a comprehensive battery of literacy assessments to identify areas of instructional need, including collecting an on-demand writing sample. The task for this writing sample varied by age and ability, but tutors were encouraged to find a writing topic that would engage their student and elicit sufficient text for analysis. For instance, younger students were offered options such as writing about their favorite animal or TV show, whereas older students might be presented with a controversy such as "should boys and girls play on the same sports teams?". Students were encouraged to brainstorm before they begin writing, and are allowed as much time as they would like to finish their writing. Most children wrote for less than 10 min, but some wrote a lot longer, others a lot shorter (particularly the younger children). Once complete, the tutors asked each child to read their composition out loud, allowing the tutor to transcribe the intended text. No corrections were made to the child's original writing. Additional writing samples were also produced periodically throughout the semester as part of their instruction and progress monitoring.

Each writing sample is transcribed digitally and annotated for potential spelling errors. For each misspelled word that was recorded (i.e., digitized), the dataset additionally includes their intended word, grade (K through 8, approximately ages 5–14), spelling development level, and the student's native language. While some of the data collected here was outside of our target age range, evaluations below were only conducted with data from children ages 6–12. For a portion of the entries (485 out of 1651) up to three words preceding the misspelling were also recorded. The spelling development levels were recorded based on established educational research (Bear et al., 1996) by analyzing features in each word to confirm their developmental level. These levels help pinpoint in which areas children are struggling with spelling to assist in further instruction. The resulting dataset consists of 1,651 entries and is referred to as $\textbf{Essay}_{\text{MSP}}$. Information regarding this dataset is compiled in Table 1; examples of dataset instances are included in Table 2.

3.2. Children's spelling errors in typed search queries

$\textbf{Essay}_{\text{MSP}}$ is comprised of written spelling samples, yet, to our knowledge, there is no dataset explicitly focused on spelling errors generated as a result of typing queries. For this reason, we conducted a number of experiments with children to enable us to collect data and build a new dataset: $\textbf{Query}_{\text{MSP}}$ (Downs, Anuyah et al., 2020; Downs, Shukla et al., 2020). Based on the protocol suggested by Landoni et al. (2019) and further specified in Downs, Anuyah et al. (2020), to elicit children's search queries, child participants were given various open-ended and fact-based prompts (list of prompts can be seen in Table 3). The child participants in these sessions used CAST (Child Adaptive Search Tool), a custom search tool, on a desktop computer for entering search queries (see Fig. 1 for picture of KidSpell in CAST). The child participants in this study were not familiar with CAST, but its design and function are similar to that of other popular search engines.

The custom search tool provides spellchecking utility that will later mark spelling errors and provide spelling suggestions. During query formulation, if a spelling error is identified, by its lack of presence in an English dictionary (Kuperman, Stadthagen-Gonzalez, & Brysbaert, 2012), it will be marked as such by being underlined and colored in red. Hovering over any spelling errors

Table 3
Sample tasks prompts given to child participants to create $\textsc{Query}_{\textsc{msp}}$.

Search task	Type
Fact-based	Who was the first computer programmer? Who was the scientist that invented robots? How far away is the Earth from the Sun? How tall is a tyrannosaurus rex? What is the state bird of Idaho? What are the first 10 digits of pi? What is the closest planet to the Sun?
Open-ended	Find me a cool fact about space. Find me a difference between Earth and Mars. Find me an interesting fact about Albert Einstein. Find me a fact about your favorite dinosaur. Find me the name of a famous mathematician. Find me a cool fact about space. Find me an interesting fact about dogs.

Table 4
Sample subset of instances in $\textsc{Query}_{\textsc{msp}}$.

Target	Spelling	Clicked	Words before
specific	pisific	pacific	was the
einstein	enistein	einstein	did albert
invented robots	evendedrobots	n/a	who

Table 5
Sexually explicit and hate-based word rate in top 5 suggestions on various spellcheckers.

	Enchant	SimSpell	Bing	Hunspell	Gingerit	Aspell
Hate-based	0.0156	0.0264	0.0029	0.0156	0.0039	0.0234
Sexually Explicit	0.0450	0.0489	0.0010	0.0421	0.0078	0.0469

provides a list of up to 5 spelling suggestions. Clicking on a spelling suggestion replaces the spelling error with the suggested spelling. All inputs made by the child participants in the interface were automatically recorded by CAST. Additionally, facilitators (made up of graduate and undergraduate researchers) observed and recorded notes based on interactions children made with the search interface with a focus on words children misspelled. As the spelling suggestion that matches the child's intended word may not always appear on the list of suggestions, and because the suggestion children selection did not always match the word they intended to spell, their intended word had to be determined based on the context of their task. The intended word for each misspelling was collectively agreed upon by four facilitators after the experiments had been completed based on query logs, search prompts given, and notes taken during sessions. They were then validated by an expert in children's literacy.

For each spelling error recorded, $\textsc{Query}_{\textsc{msp}}$ includes the spelling suggestion that was clicked on, the agreed-upon intended word, up to three words before the spelling error, and the user's session ID. The resulting dataset (summarized in Table 1) includes 134 entries, samples of which can be seen in Table 4.

4. Spelling correction algorithm

In this section, we describe our English, child-oriented spelling correction method. We first discuss the method for candidate generation using a phonetic encoding algorithm tailored towards children, termed **KidSpell**, which is illustrated in Fig. 1. This method returns potential candidates in order of term frequency. We improve on the ranking of candidates based on a number of features using the lambdaMART Learning to Rank model. We term the full model **KidSpell**λ. The architecture of the model is illustrated in Fig. 2.

4.1. Phonetic candidate generation

The goal of the candidate generation is to reduce the search space such that a spelling correction method can efficiently rank the given candidates rather than observing and processing every word in the dictionary. As children have a tendency to use phonological strategies over orthographic ones (Greenberg et al., 2002), spelling correction methods that use a phonological approach show promise. As such, we take inspiration from Deorowicz and Ciura (2005) who showed that phonetic similarity strategies are effective in correcting spelling errors made by those who know the pronunciation, but not the spelling, which is a common error made by children. At a high level, given a misspelled term written by a child, our candidate generation model applies a phonetic similarity approach in order to identify potential spelling candidates.

4.1.1. Dictionary creation

Critical to generating candidates is a list of valid words, known as a *dictionary*. We created a child-friendly dictionary that is lacking in typical spellcheckers. To achieve this we derive our dictionary from age of acquisition research which identifies the typical age words are learned (Kuperman et al., 2012).

A child-friendly spellchecker should also refrain from producing any sexually explicit, hate-based, or other inappropriate words. An investigation into the extent to which spellcheckers produce spelling suggestions that include sexually explicit or hate-based words showed many popular spellcheckers have a tendency to produce such words in up to 5% of their suggestions (Downs, Anuyah et al., 2020) (See Table 5). For this reason, we ensure that inappropriate words are excluded from the aforementioned child-friendly dictionary. We considered 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.[2] Hate-based words were identified as those that exist among the list of hate-speech and offensive language lexicons which we compiled from HateBase,[3] a repository of hate-speech language. It is worth mentioning limitations related to excluding such words from the dictionary. Some of the explicit terms included among the Google and HateBase bad word list are ambiguous in nature and may not necessarily be inappropriate when considered in certain cultural or educational contexts, e.g., *screw* and *slave*. In the classroom context, we perceived that preventing children's exposure to a potential false positive (i.e., a word that may be relevant to the classroom in one sense but is flagged as it is inappropriate in another sense) is less harmful than providing said word and potentially leading to the retrieval of inappropriate resources for children. Hence, we discarded from our dictionary all of the terms that exist in the sexually explicit and hate-based word lists.

In total, our dictionary is comprised of 60,847 unique words. Some spelling correction methods, intended for general audiences, use lexicons of up to 1.2 million words (Li et al., 2011). Considering that children typically acquire around 60 thousand words in their first 18 years of life (Bloom, 2002), our limited lexicon of child-known word is intended to assist in providing more appropriate spelling suggestions.

4.1.2. Phonetic encoding approach

Our approach to finding phonetically similar candidates is comparable to the one found in the SoundEx model described by Croft et al. (2010): words are encoded to produce a phonetic key that groups words together with ones that are similarly

[2] https://code.google.com/archive/p/badwordslist/.
[3] https://www.hatebase.org/.

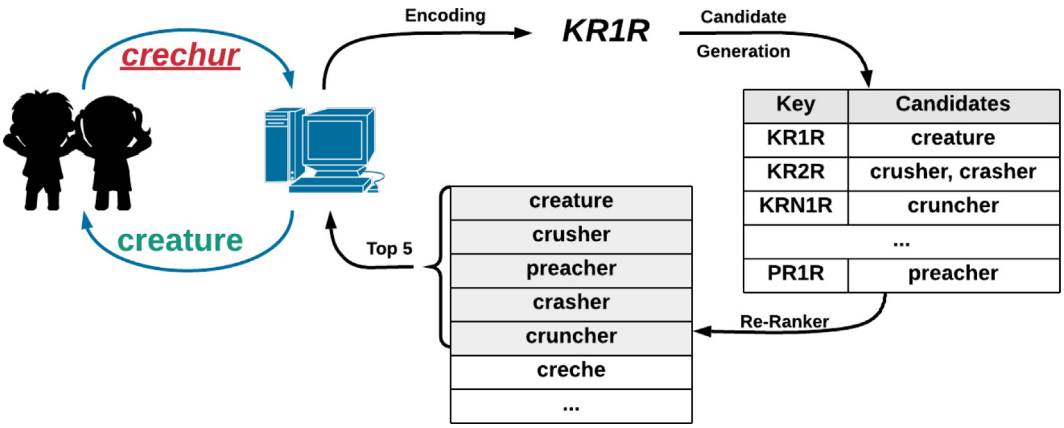


Fig. 2. KidSpellλ architecture using the spelling error *crechur* for the word *creature*.

pronounced. However, our phonetic key encoding takes inspiration from the Metaphone algorithm (Philips, 1990) to produce smaller groupings with less general phonetic representations. For example, our encoding differs in that the letters F and V are not considered the same sound resulting in words like *fan* and *van* being encoded to different phonetic keys. Similar is the case with Q and K, resulting in *quail* and *kale* being encoded to different keys. Although those letter pairs 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. In general, the phonetic encoding includes common phonetic rules, such as recognizing the letter sequence *ph* makes the /f/ sound and the *k* in letter sequences 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 key groupings that are too small, resulting in several keys that only match a single word. The full phonetic ruleset for our encoding is described in Table 6. Words and misspellings are modified by each rule in the ruleset in the order shown and the result is the final phonetic form.

To illustrate the phonetic encoding process, as pictured in Fig. 2, consider the word *creature* which would be processed as follows:

- *t* in *ture* makes the *CH* sound (encoded as 1), transforming the word to *crea1ure*
- *c* is not followed by *i*, *e*, or *y*, so it makes the *K* sound, resulting in *Krea1ure*
- Remove vowels resulting in the final phonetic form: ***KR1R***

As a pre-computational step, this process is performed on every word in the previously described dictionary and a mapping is made 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*.

To produce a larger assortment of spelling correction candidates we then use Levenshtein distance (Levenshtein, 1966) on the key of the given misspelled word to find similar keys. Levenshtein distance is the minimum number of character insertions, deletions, and substitutions needed to transform one word to another. For example, given the misspelled word *talbe* (intended to be *table*), we would take the encoding of the misspelling, ***TLB***, and generate encodings that are 1 edit distance apart (***TBL***, ***TLBR***, ***DLB***, etc.). Despite that the intended word *table* has a different key (***TBL***) than the misspelled word, this allows to quickly find it and add to the pool of candidates. If the amount of words produced by keys of 1 edit distance from the original does not meet the number of requested words, we then generate keys at an increasing edit distance until the number of requested words is met. Candidates are returned in ascending order of their edit distance between the keys and secondarily by their frequency in the Simple English Wikipedia,[4] with more frequent words being ranked higher.

4.2. Candidate ranking

Once we have a suitable list of spelling correction candidates, it is essential that we rank them appropriately such that the intended word is positioned towards the top of the list. The phonetic key of spelling errors may match several different words and it is possible the key of the intended word does not match the spelling error. Children have also shown to have a propensity to interact with higher-ranked alternatives (Anuyah et al., 2018; Gwizdka & Bilal, 2017). For these reasons, we re-rank our spelling suggestion candidates with a candidate ranking strategy based on several informative features.

The candidate ranking strategy is inspired by the work of Fomin and Bondarenko (2018). In similar work, edit distances have been used in probabilistic and machine learning methods to determine word similarity (Brill & Moore, 2000; De Amorim & Zampieri, 2013; Huang et al., 2013, 2013). However, it has been shown that edit distance methods are ineffective at correcting children's spelling errors (Downs, Anuyah et al., 2020). Phonetic similarity techniques, on the other hand, are more capable of correcting the spelling errors children make (Deorowicz & Ciura, 2005; Downs, Anuyah et al., 2020) and as such we use a features set that instead considers phonetic similarity to improve their effectiveness.

We create a feature extractor that takes in two strings: the original incorrectly spelled word and the suggested spelling. It then returns the following features which are then used to rank spelling suggestion candidates:

1. The difference in length between the suggestion and the misspelling. This featured showed promise in Fomin and Bondarenko (2018).
2. Levenshtein distance between the suggestion and misspelling. This is a traditional spellchecking strategy.
3. Frequency of the suggestion in Simple Wikipedia articles. Word frequency has been shown to be an effective method for ranking spelling suggestions (Mitton, 2009).
4. An n-gram (contiguous words) language model inferred from the frequency and sequences of words found in simple Wikipedia articles. (Specifically we utilized an interpolated Kneser–Ney n-gram model which has shown to

[4] Simple English Wikipedia: https://simple.wikipedia.org/.

Table 6
Ruleset used to transform a word or spelling error into a phonetic key. Common word endings consist of *s*, *ing*, *ings*, and *ed*.

Step	Rule
1	Convert 'cc' to 'K'
2	Replace consecutive duplicate consonants with a single consonant
3	Convert 'ck' to 'K'
4	Convert 'ocea' at the start of a word to 'A2'
5	Convert vowels at the start of a word to 'A'
6	Convert 'gn', 'kn', or 'pn' at the start of a word to 'N'
7	Convert 'wr' at the start of a word to 'R'
8	Convert 'x' at the start of a word to 'S'
9	Convert 'wh' at the start of a word to 'W'
10	Convert 'gh' at the start of a word to 'G'
11	Convert 'rh' at the start of a word to 'R'
12	Convert 'sch' at the start of a word to 'SK'
13	Convert 'y' at the start of a word to 'Y'
14	Convert 'mb' at the end of a word or before a common word ending to 'M'
15	Convert 'th' to '0'
16	Convert 'ch' or 'tch' to '1'
17	Convert the t in 'ture' or 'tual' to '1'
18	Convert 'sh' to '2'
19	Convert the 'c' in 'cion' or 'ciou' to '2'
20	Convert the 't' in 'tian', 'tion', or 'tious' to '2'
21	Convert the 's' in 'sian', 'sion', or 'sious' to '2'
22	Convert the 'c' in 'ci', 'ce', 'cy', 'sci', 'sce', or 'scy' to 'S'
23	Convert remaining 'c' to 'K'
24	Convert 'dge' to 'J'
25	Remove 'gh' if the next letter is a consonant
26	Remove 'gh' at the end of a word or before common word ending
27	Convert remaining 'gh' to 'G'
28	Convert 'gn' at the end of a word or before a common word ending to 'N'
29	Convert 'y' at the end of a word to 'Y'
30	Convert 'ph' to 'F'
31	Remove 'h' if before vowel, end of word, or common word endings
32	Remove 'w' if before consonant, end of word, or common word endings
33	Convert 'z' to 'S'
34	Remove remaining vowels, convert remaining consonants to capital

be an effective smoothing technique (Goodman, 2001).) The n-gram language models were shown to be a highly important feature for spellchecking in the work presented in Fomin and Bondarenko (2018).

5. Age of Acquisition (AoA). AoA research provides us with the age words are typically learned which is likely important when providing words for children (Kuperman et al., 2012).
6. Levenshtein distance between the phonetic codes of the suggestion and misspelling as determined by the KidSpell phonetic algorithm. This can tell us how similar words are phonetically.
7. Levenshtein distance between the phonetic codes of the suggestion and misspelling as determined by the SoundEx phonetic algorithm. Another phonetic algorithm inspired from the work in Fomin and Bondarenko (2018).
8. Whether or not the first letter of the KidSpell phonetic codes match between the suggestion and the misspelling. Some spellcheckers often assume the first letter in a misspelling is correct, which it often is (Mitton, 2009), this instead looks at the first sound.
9. Number of corrections where a letter has an incorrect number of consecutive repetitions (e.g., *ammmmaaaaazing* → *amazing*: 2 corrections). This is a type of error children are known to make (Dragovic, Madrazo Azpiazu, & Pera, 2016).
10. Number of unique consonants (i.e., number of consonants that appear either the suggestion or misspelling, but not both). Previous findings have shown that vowels were not necessary when identifying the correct spelling from a given misspelling (Downs, Anuyah et al., 2020).
11. Number of unique vowels between the suggestion and the misspelling. In contrast to the feature above, this looks for the similarity between vowels used.

Up to 50 candidates are generated for each misspelling then features are extracted on each. After features are extracted, we follow a similar approach to the one described in Fomin and Bondarenko (2018). For each misspelling, we have up to 1 possible correct suggestion (assigned the value 1) and many incorrect suggestions (assigned the value 0). These are then transmitted to a machine learning model. Given that providing suitable spelling suggestions in an interactive spellchecker is truly a ranking problem rather than a classification problem, we use the learning-to-rank (LTR) model LambdaMART (Burges, 2010). While the use of LTR models has not knowingly been explored for spellchecking, they have proven effective at similar ranking problems such as large scale search, query suggestions, and recommendation (Santos, Macdonald, & Ounis, 2013). We empirically verified that the lambdaMART LTR consistently outperformed other machine learning models (evaluations and details on the machine learning models are reported in Appendix). As such LambdaMART is the re-ranking model used in KidSpellλ.

4.2.1. LambdaMART training

The LambdaMART model is trained to maximize on a specific metric and for this we use Mean Reciprocal Rank (MRR) with a max position of 5. MRR is calculated as defined in Eq. (1).

$$MRR = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{1}{rank_i} \tag{1}$$

where C is the set of spelling errors, $|C|$ is the number spelling errors, and $rank_i$ is the ranking position of the gold standard. As there is only 1 correct suggestion amongst many suggestions, MRR is a suitable metric for this task. A max position of 5 gives no value to items ranked outside the top 5, which prioritizes ranking suggestions within at least the top 5. Evaluations commonly look

at either the top 5 or top 10 suggestions returned by spellcheckers. Given that a child audience must be considered and children tend to favor higher-ranked alternatives (Anuyah et al., 2018; Downs, Shukla et al., 2020), we favor placing suggestions within the top 5. Using the CHILDRENS$_{MSP}$ dataset, K-Fold cross validation and grid search were used to find the best hyperparameters for LambdaMART, which are listed as follows:

Max Depth: 5
Learning Rate: .1
Estimators: 50
Minimum Split Samples: 2
Minimum Leaf Samples: 1

The LambdaMART model accepts a relevancy or target value for each set of items for the training process. For this task, the one correct suggestion is given a value of 1 and all other incorrect suggestions are given a value of 0. We refer to this improved ranking model as KidSpellλ.

5. Spelling correction evaluation

In this section, we present an evaluation of the effectiveness of our method in two areas: the first being the *candidate generation* of KidSpell and the second being *candidate ranking* of KidSpellλ. Both are evaluated using cross-validation on spelling errors generated by children described in Section 3. We also examine the relative feature importance for the features described in Section 4.2.

5.1. Candidate generation

The experiments in this section examine both the effectiveness and the efficiency of KidSpell's candidate generation method against state-of-the-art methods. Each candidate generation method (KidSpell's and the baselines) is asked to generate candidates for each spelling error in the CHILDRENS$_{MSP}$ dataset. We vary the number of k candidates generated from each method because they directly influence the time complexity and the search space reduction. This evaluation method is used in other candidate generation research (Pande, 2017).

Baselines. The most similar work to ours for generating suitable candidates is the work done by Pande (2017). Pande utilizes neural character embeddings that employ character sequences that are generated using consecutive vowels or consonants, but not both (e.g., 'affiliates' generates 'a ff i l ia t e s'). Our implementation uses size 100 embeddings as that provided the best performance. We also used a modified version of their algorithm that instead employs single character sequences used in character embeddings, which performed better with children's misspellings (e.g., 'affiliates' generates 'a f f i l i a t e s'). All methods utilized the same dictionary for candidate generation.

Metrics. We report on the *success rate* which is the percentage of spelling errors for which the gold standard (intended word) is among the pool of suggestions generated. As the goal of these methods is to reduce the time complexity of spelling correction algorithms by limiting the search space, we also measure the runtime (in seconds) to generate k number of candidates for all words in the entire CHILDRENS$_{MSP}$ dataset.

Results. The success rates of the *KidSpell* phonetic algorithm and the two baselines (labeled *Pande Embeddings* and *Character Embeddings*) are presented in Fig. 3. The KidSpell phonetic algorithm significantly outperforms the two baselines for every variation of k candidates using a paired t-test ($p < 0.05$; n=1785). Most noticeably it outperforms significantly on the lower end of k variations. Just generating 100 candidates with the *KidSpell* phonetic algorithm outperforms both baselines even when given the opportunity to generate 5000 candidates. Notably, the embeddings methods had difficulties picking up spelling errors that were more than just a couple edits away (e.g., *favtit* for *favorite* requires 3 single character edits). They also tended to return substrings of the misspelling that matched a real word (e.g., returning *since* for the misspelling *sincerly*). The KidSpell phonetic algorithm benefited from returning words in order of frequency as the embeddings had a tendency to return obscure words.

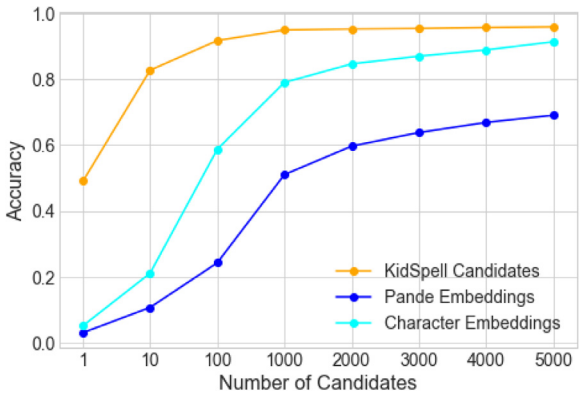


Fig. 3. Success rates (%) for various k (number of candidates).

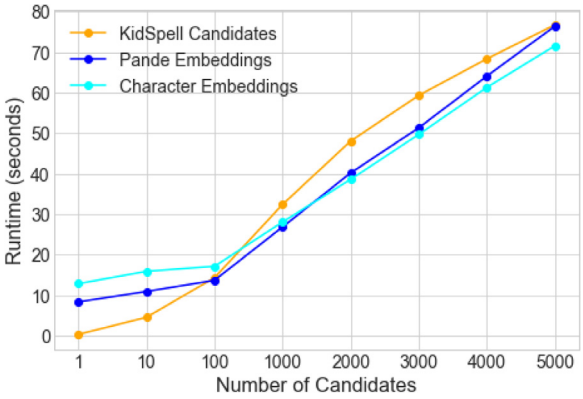
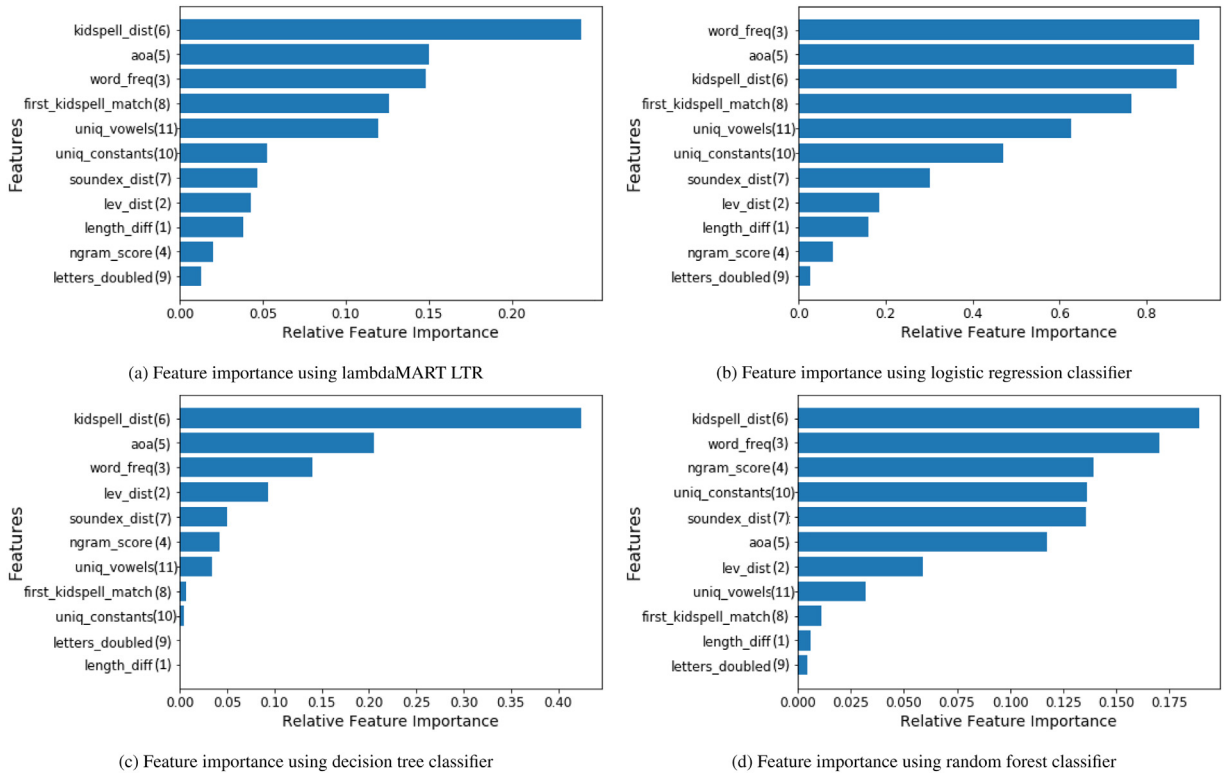


Fig. 4. Runtime in seconds for various k (number of candidates).

The runtime of the various algorithms are reported in Fig. 4. The *KidSpell* phonetic algorithm is more efficient when generating a lesser number of candidates (<100), but falls slightly behind when generating 1000 or more candidates. When combined with the findings based on success rates, the KidSpell phonetic algorithm can effectively find a more precise and smaller candidate pool in a fraction of the time. For example, 100 KidSpell candidates is more likely to contain the correct suggestion than any of the baselines at 5000 candidates and can be generated in just a sixth of the time. In fact, generating more than 100 candidates using KidSpell's phonetic algorithm becomes unnecessary as we nearly achieve our peak performance.

5.2. Feature importance

The features importance for each of the features listed in Section 4.2 for the lambdaMART LTR model are listed in Fig. 5(a). We also include the feature importance for other tested models: logistic regression, decision tree, and random forest classifiers are reported in Figs. 5(b), 5(c), and 5(d) respectively. The higher the value, the more important the feature is. In all but the logistic



(a) Feature importance using lambdaMART LTR

(b) Feature importance using logistic regression classifier

(c) Feature importance using decision tree classifier

(d) Feature importance using random forest classifier

Fig. 5. Feature importance each machine learning model. Each feature is numbered corresponding with the enumeration in Section 4.2.

regression, the edit distance of the KidSpell phonetic algorithm is considered the most important feature. For the lambdaMART LTR, it is by far the important component of the feature set. This emphasizes the importance of phonetic information when correcting children's spelling errors. Edit distance of the SoundEx phonetic algorithm on the other hand scores low. While they did not have a high correlation, they do fill similar rolls and the KidSpell phonetic algorithm may provide much more precise information.

Age of acquisition and word frequency are second and third in feature importance when used in the lambdaMART in KidSpellλ, and are found to be important features for the other models as well. Given that children are more likely to know or use more frequent words or words within their age of acquisition (Kuperman et al., 2012), it is unsurprising that the models value these two features. The unique number of vowels is another highly important feature for lambdaMART. This feature is surprising as the KidSpell phonetic algorithm was built around children's inconsistent use of vowels and ignores them altogether. Perhaps, the lack of information of vowel usage that we are getting from the KidSpell phonetic edit distance puts a higher value on this feature. N-gram scores have been important in other work that attempted to correct adult spelling (Fomin & Bondarenko, 2018), but scored low with most of our models.

5.3. Spelling correction

In this section, we evaluate the full KidSpellλ spellchecking model using the improved ranking described in Section 4.2, against other state-of-the-art spellcheckers. We examine the performance of KidSpellλ and the baseline counterparts introduced below using the CHILDRENS$_{MSP}$ dataset described in Section 3. This includes both the spelling errors made in hand-written essays as well as typed search queries. For models that require training (such as the KidSpellλ model), reported results are the average from a 5-fold cross-validation (80% training, 20% test).

Baselines. Several baselines were chosen to compare to our method, these include:

- **Aspell (Normal)** - Aspell was chosen as a common spellchecking baseline (Brill & Moore, 2000; Deorowicz & Ciura, 2005). Aspell also utilizes phonetic encodings (Metaphone algorithm (Philips, 1990)) in a similar manner to our approach making it potentially effective for children's spelling errors.
- **Aspell (Bad Spellers)** - Aspell with *Bad Spellers* mode enabled was chosen as the goal of correcting the spelling of bad spellers is the most similar to our work.
- **Bing** - Microsoft's Bing Spell Check API was used as an industry standard for correcting spelling in the search context. Further, as stated by Bilal and Boehm (2017) "children hardly use search engines designed for their age levels", so we use this as one of the spellcheckers young searchers could encounter as they often use large scale search engines (e.g., Google and Bing) (Bilal & Boehm, 2017).
- **KidSpell** - The final baseline is the KidSpell candidate generation method without the improved ranking as described in Section 4.2.

Each baseline uses the dictionary supplied with the software. Both versions of KidSpell use the same dictionary.

Metrics. To measure the performance of the respective spellcheckers we use Hit Rate and MRR.

Hit Rate measures the rate at which the known intended word (i.e., gold standard) appears in the list of spelling suggestions. For each spelling error in the dataset, the spellchecker receives a value of 1 if the gold standard is in the list of spelling suggestions, otherwise, this value is 0. An average is taken to determine Hit Rate.

MRR measures how well ranked the spelling suggestions are by capturing the average position of the relevant spelling suggestion. A higher MRR value indicates that overall the gold standard (i.e., desired spelling correction) is positioned higher in the

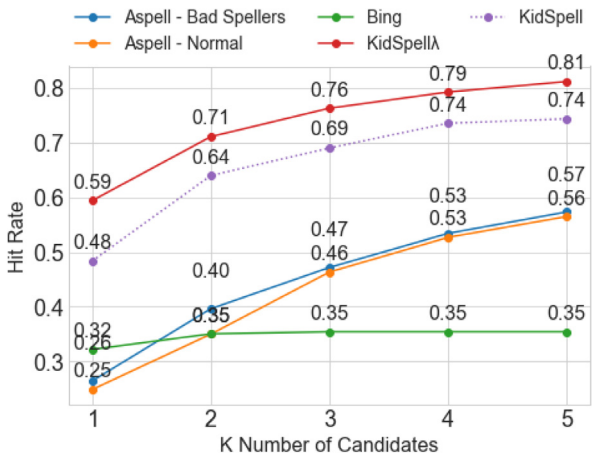


Fig. 6. Hit-Rate@k (where k is the number of provided suggestions) on the CHILDRENS$_{MSP}$ dataset. Note that the *Bing Spell Check API* only returns a maximum of 3 suggestions.

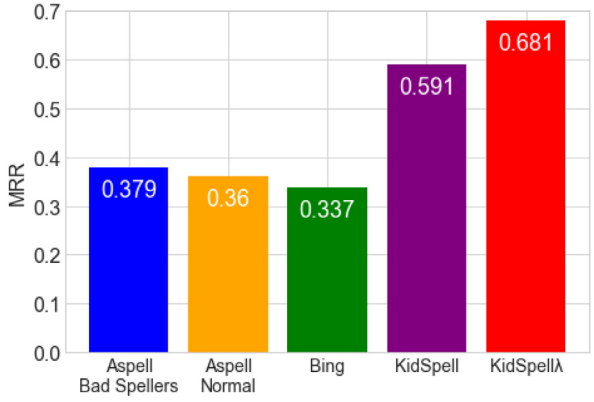


Fig. 7. MRR using top 5 suggestions on the CHILDRENS$_{MSP}$ dataset.

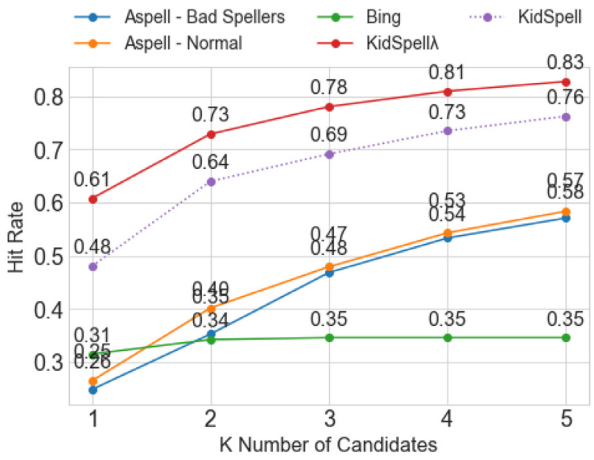


Fig. 8. Hit-Rate for various k (number of suggestions) on spelling errors made in hand-written essays. Note that the *Bing Spell Check API* only returns a maximum of 3 suggestions.

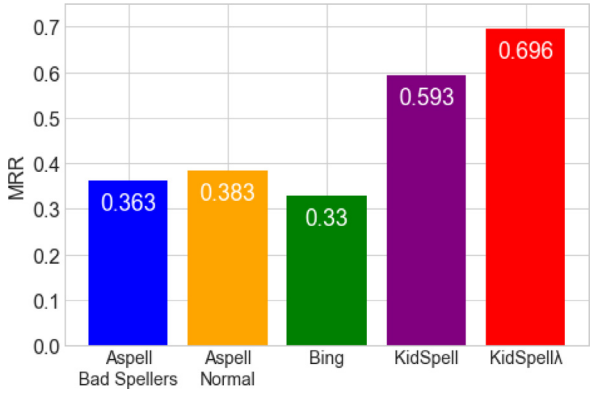


Fig. 9. MRR using the top 5 suggestions for hand-written essays.

ranking of the suggestions. Given children's propensity to click on higher-ranked alternatives for spelling suggestions (Downs, Shukla et al., 2020) as well as other areas of search (Anuyah et al., 2018; Gwizdka & Bilal, 2017), it is crucial to rank the gold standard highly. Taken together, these two metrics measure how well each spellchecker is at finding the intended word (Hit Rate) and how well it is ranked (MRR). Numbers for hit rate are reported for a varying number of spelling suggestions (k 1–5) and MRR is reported using the top 5 suggestions.

Results. Results of the hit rate of KidSpellλ along with the baselines on the full CHILDRENS$_{MSP}$ dataset are described in Fig. 6. Note that the *Bing Spell Check API* only returns a maximum of 3 suggestions per spelling error, limiting its performance when k is greater than 3. The reported KidSpellλ results are significantly better when compared to the baselines intended for adult users as well as the original KidSpell method using a paired t-test ($p < 0.05$; n=1785). While Aspell's Bad Spellers mode does provide a minor increase over the normal Aspell, it still has difficulties when dealing with children's spelling errors. Even just providing 1 suggestion from KidSpellλ is more likely to provide the gold standard than 5 suggestions from the best alternative (Aspell Bad Spellers mode).

Results for the MRR scores for each spellchecker are provided in Fig. 7. The improvement for KidSpellλ is statistically significant using a paired t-test over all alternatives ($p < 0.05$; n=1785). These scores indicate that KidSpellλ is able to include the gold standard within the first 2 suggestions on average, while the alternatives provide the gold standard at the 3rd position on average.

We further examine the results of the spellcheckers on the two different environments they were created in (hand-written essays and typed search queries). The hit-rate and MRR for spelling errors made in hand-written essays are reported in Figs. 8 and 9 respectively. Given that the large majority of the samples in CHILDRENS$_{MSP}$ is made of hand-written essay spelling errors (1651 out of 1785) we see similar results to those of the full dataset. Likewise, the improvement for KidSpellλ is statistically significant for both the hit-rate and MRR ($p < 0.05$; n=1651).

The hit-rate and MRR for spelling errors made in the typed search queries are reported in Figs. 10 and 11 respectively. Here, all but Bing Spell Check perform considerably worse than on the hand-written essay spelling errors. We attribute this to the *mistyping* errors that can occur while using a keyboard. Since KidSpellλ and Aspell both rely on phonetic information, mistyping errors such as *tghat* for *that* make the words seem like unlikely matches since the misspelling and the gold standard do not match phonetically. Similarly, typed spelling errors are much more likely to include *boundary errors*. These errors consist of including a space when there should not be one (e.g., *com puter* for *computer*) or the lack of a space when there should be one (e.g., boisestate for *boise state*). These types of errors are overlooked by KidSpellλ. Bing Spell Check is better at handling these types of errors and has improved performance when

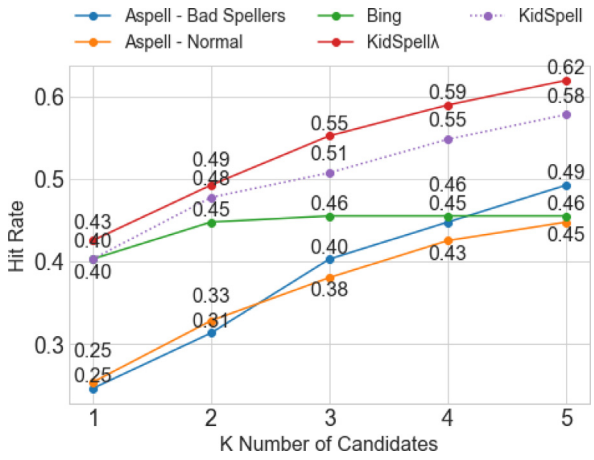


Fig. 10. Hit-Rate for various k (number of suggestions) on spelling errors made in typed search queries. Note that the *Bing Spell Check API* only returns a maximum of 3 suggestions.

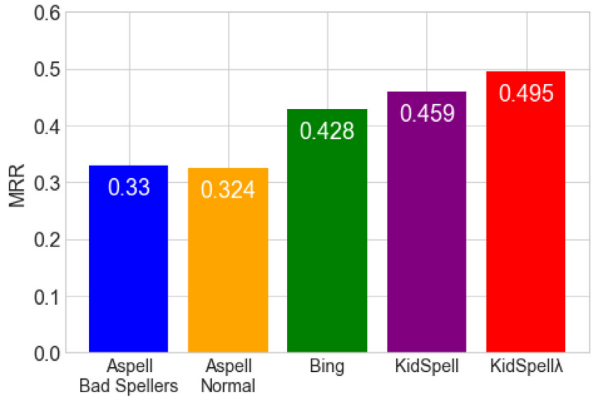


Fig. 11. MRR using top 5 suggestions for typed search queries.

being used for its intended purpose (i.e., correcting misspelled queries) however it is still significantly worse in performance when compared to KidSpellλ. KidSpellλ's improvement over both the original method and the alternative is statistically significant when providing 5 suggestions using the paired t-test ($p < 0.05$; n=134). It is also worth noting that although KidSpellλ was trained on primarily hand-written essay spelling errors, it still provides a significant improvement over the original KidSpell method when handling typed query errors.

5.4. Grade levels, spelling levels, and native languages

To further analyze KidSpellλ when compared to other spellcheckers, we evaluate their capability to correct spelling errors from children of different grades, spelling abilities, and native languages, which are included as part of the hand-written essay spelling errors as described in Section 3. As such, all spelling errors in this section are all a subset of the hand-written essay spelling errors dataset ($\textbf{ESSAY}_{\text{MSP}}$).

Evaluations on the hit-rate at 5 of the spellcheckers on children's spelling errors, grades K through 8, are included in Fig. 12. While most spellcheckers see a general trend upward in hit-rate as children's grade level increases, KidSpellλ experiences the least variation while maintaining a higher hit-rate at all grade levels. The dip for most spellcheckers at grade 7 can be explained by the limited data and number of children at that grade level. The

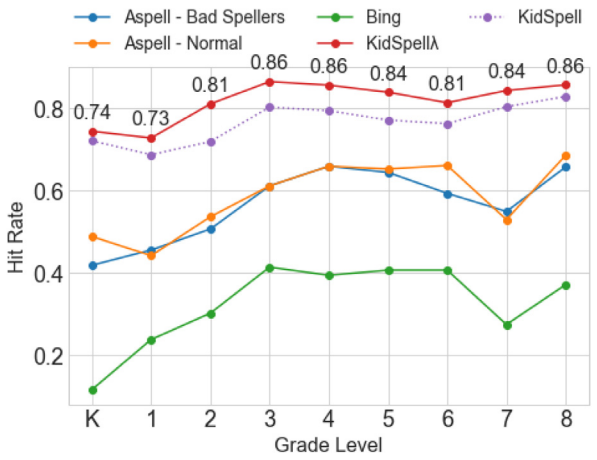


Fig. 12. Hit-Rate at 5 for spelling errors made by children in grades K through 8.

Table 7
Number of spelling errors for each Grade Level Kindergarten (K) through 8.

Grade	K	1	2	3	4	5	6	7	8
Ages[a]	5–6	6–7	7–8	8–9	9–10	10–11	11–12	12–13	13-14
Errors	43	147	470	355	355	118	59	51	35
Children	6	15	29	22	16	9	5	4	4

[a]Ages are approximate as the data identified only the grade; these are the usual ages of children in each of these grades although sometimes ages differ.

Table 8
Number of spelling errors for each spelling development level from the $\textbf{ESSAY}_{\text{MSP}}$ dataset.

Level	Level 1	Level 2	Level 3	Level 4	Level 5
Errors	5	546	827	186	26
Children	2	25	45	16	6

number of spelling errors and children for each grade level are included in Table 7.

As grade level is not necessarily indicative of a child's spelling ability, we also include evaluations on the hit-rate of the various spellcheckers separated by the spelling developmental level of the children as described in Section 3. The hit-rate at 5 is included in Fig. 14. A table of the number of spelling errors and children for each development level is included in Table 8. Similar to the grade levels, spellcheckers see a trend upward as spelling level increases. This is especially noticeable for the adult-oriented baselines. The closer the spelling level of the student gets to the intended audience (i.e., adults), the better they perform. Meanwhile, KidSpellλ is able to keep a hit rate of 80% or higher regardless of spelling development level. The results seen for the different grade levels and spelling levels emphasize KidSpellλ's importance for especially young or new spellers. The drop in hit-rate at the highest spelling level and inconsistencies for the lowest spelling level can be explained by the limited amount of data for each of those groups.

For a partial amount of the hand-written data, children's native language was recorded. We examine the hit-rate at 5 of the different spellcheckers for English and non-English speakers in Fig. 13. As the phonetic information for both KidSpellλ and Aspell are based in English we might expect that they perform worse when handling spelling errors made by children that speak a different native language. While KidSpellλ performs slightly better when handling non-English speakers and Aspell performs slightly worse, the differences were not significant (2 sample t-test; $p > 0.05$). Regardless of the native language of the children,

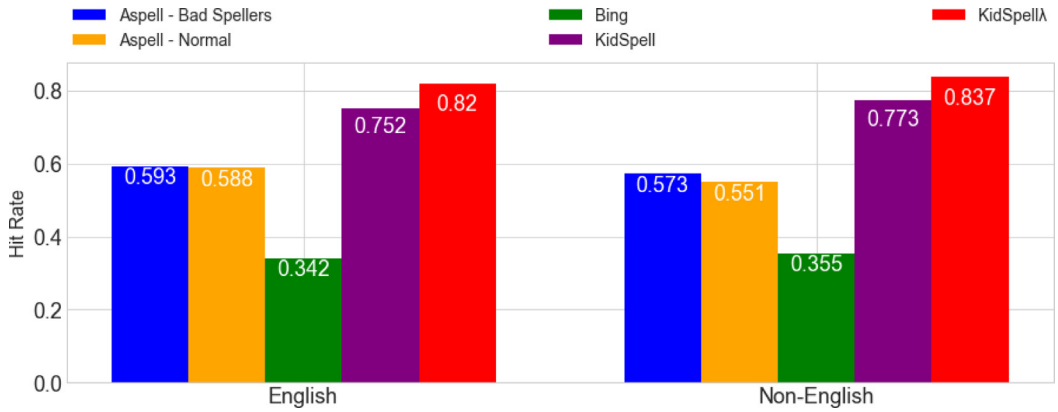


Fig. 13. Hit-Rate at 5 for spelling errors made by children with English and non-English native languages.

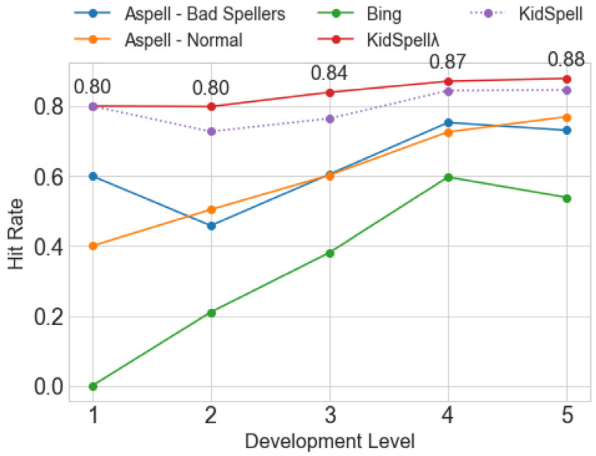


Fig. 14. Hit-Rate at 5 for spelling errors made by children at different spelling developmental levels.

KidSpellλ is consistent in achieving a hit-rate of at least 80% and significantly outperforms baselines (paired t-test; $p < 0.05$).

5.5. Discussion

In Sections 5.1 through 5.4 , we described the experiments conducted to assess the performance of our base KidSpell candidate generation as well as the improved KidSpellλ that ranked the candidate suggestions. We found that candidate generation based on children's phonetic spelling behaviors was significantly better at finding suitable spelling candidates. This aligns with children's spelling behavior research by Greenberg et al. (2002). In fact, using an improved candidate generation method alone was often more successful than the baseline spellcheckers (Aspell and Bing).

When it comes to feature importance in the machine learning tasks (meaning how much it informed the ranking of suggestions), we found that the two most highly valued features were: (1) the phoenetic similarity between the mispelled word and the intended word, and (2) the age of acquisition for the intended word. The value of phonetic similarity was not surprising as this once again directly relates to children's spelling behaviors Greenberg et al. (2002). The importance of age of acquisition was also unsurprising as this considers words children are more likely to know (Kuperman et al., 2012). N-gram scores were relatively important. This is an interesting revelation in children's spelling as related work using n-gram scores for adult spellings were considered highly important (Fomin & Bondarenko, 2018). Unsurprisingly, Levenshtein distance was typically unimportant. This aligns with our findings that spellcheckers that use Levenshtein edit distance on the characters that are typed do not perform well when correcting children's spelling errors (Downs, Anuyah et al., 2020).

Regardless of the child's grade level, development level, or native language, KidSpellλ performed better than the baseline spellcheckers in recommending the intended word. Spellcheckers had more success the higher grade level or the higher the spelling development level of the child was. These findings are comparable to the findings by Figueredo (2006) that did not find a difference in children's success using a spellchecker between children of grade 4 and 6. Comparing success between native (English) and non-native speakers, KidSpell provided minimal advantages for non-native speakers. This is somewhat unexpected as KidSpell makes use of English-based phonetic similarity methods.

The dataset used to evaluate had 1785 samples. While a larger sample is always welcomed, the results of our analyses indicate that are results are statistically significant. We believe this is statistically meaningful as using this sample of real children's misspellings, the suggestion algorithm is better for children's spelling errors and the cues make a significant difference with regards to encouraging users to explore and select the intended word.

6. Spellchecking interface

Despite advances demonstrated in spelling correction for children, we cannot expect algorithms to fully recover the intent of the child as the first spelling suggestion, which is a reason why spellcheckers provide multiple suggestions. In the rest of this section, we discuss the visual and audio cues explored to help children identify and select their intended spelling suggestions. We then discuss the findings from our participatory design sessions in an effort to form potential solutions to known issues with spellchecking interfaces, give insights on children's views of spellchecking interfaces, and how to improve them going forward.

6.1. Multimodal cues for spelling suggestions

Dual Coding Theory posits that providing information in multiple modalities aids readers comprehension (Sadoski et al., 1993) which aligns with research that has shown children's preference for multiple inputs (Druin et al., 2010; Gossen et al., 2012). This led to investigating the effect multimodal cues (i.e., images and audio playback) can have on assisting children to select the word that best matches their intent from a list of spelling suggestions (Downs, Shukla et al., 2020). While a child-oriented spellchecker may better respond to children's spelling errors,

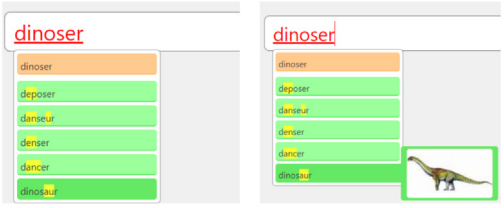


Fig. 15. Spellchecking function without visual aids (left) and with visual aids (right). Both may be accompanied by audio playback.

children's behavior to gravitate towards higher-ranked spelling suggestions (Downs, Anuyah et al., 2020), even if the word does not match their intent, can impede its effectiveness. To enhance spellchecking functionality, we evaluated spellchecking interfaces that incorporated different multimodal cues on how they affected children's selection habits and if they can help children effectively select the spelling suggestion that meets their original intent.

In this new study, we used a between-subject experimental design using a custom search tool with four conditions, each with varying media cues to go along with the spelling suggestions: no cues, audio, image, or both audio and image. There were 191 child participants (age 6–12) who took part in different experiments located at local STEM events. Each child interacted with a custom search tool with one of the conditions chosen randomly.

In the custom search tool, when a word is identified as potentially misspelled, it is colored and underlined in red. Hovering over the misspelled terms opens a list of spelling suggestions. When a spelling suggestion is hovered, an image is displayed and/or speech synthesis is used to read the word aloud depending on the condition. Images used in the interface were acquired using Google's Image Search API with safe search enabled. The first image returned, using the spelling suggestion as the search query, was the image chosen to be displayed alongside each spelling suggestion. Speech synthesis for the audio playback is acquired using Amazon Polly.[5] Examples of the different visual interfaces can be seen in Fig. 15.

In order to identify which multimodal cues, if any, would better guide children's ability to select their intended word, we adapted the protocol defined in Landoni et al. (2019) to allow us to systematically compare across separate experiments. The protocol outlines four dimensions which we specify as follows:

1. **Task**. Verbal prompts were given to child participants to serve as a starting point for a typical online search query. For this, we relied on two types of prompts: *fact-based*, which are less complex and require children to locate specific and quick answers, as well as *open-ended* prompts, which require more in-depth consideration of the content of the search results or may require multiple searches. The prompts were the same ones utilized to gather data in our initial search task (see search prompts in Table 3).
2. **User group**. Participants were children ages 6 to 12 (n = 191). We selected this age group to represent children who have likely developed the basic phonetic skills needed to attempt spelling, but have yet to obtain advanced orthographic skills (Bear et al., 1996).
3. **Strategy**. We use a custom search tool (CAST) in which all user inputs with the interface are automatically recorded. For most children (a few had seen it before), this was their first time interacting with CAST. Facilitators observed and recorded interactions children made with the search tool. Each child was accompanied by a single facilitator and up to six different facilitators were used. The interactions were automatically recorded by CAST and the notes record by facilitators were used in combination to analyze children's interactions with the interface. Each child was given 1–3 prompts and, depending on their search skills, spent less than minute or up to several minutes on a prompt. No child participated in a session more than once. Three sessions were conducted on separate days.
4. **Environment**. Search tasks were performed by children at local STEM events hosted at three local venues — two elementary schools and a local community building. The STEM event held at the community building was organized by state government agencies where children were bussed in from their respective schools and participated as an informal education (i.e., a field trip) experience. Each event contained multiple STEM-related booths and hands-on activities. Headphones were provided to assist with focus and the audio cues.

In our experiments, we examined how accurately children clicked on their intended word and in which position those clicked occurred. The results of these experiments are summarized in Table 9. We saw significant improvements to children's ability to find their intended word among a list of spelling suggestions when using either of the multimedia cues or a combination of the two. Although these improvements were shown to be of statistical significance when compared to the baseline experiment that had no cues (two-proportions z-test; $p < 0.05$), we did not observe a statistical significance when comparing them to each other (two-proportions z-test with Bonferonni correction; $p > 0.016$).

The audio only condition performed the best with the highest accurate click percentage (92%) as well as having 0 incorrect clicks in the first position and 3 incorrect clicks in the first three positions, which was half or less than any of the other experiments. Other conditions noticeably still had children resort to clicking on suggestions in the first position, even if that was not their intended word. Given that the condition with both audio and images performed worse than the audio condition suggests to us that the use of images may have had a direct impact on how useful the audio was. Why images did not have as much as an impact is an open question, but this could be explained by the fact that spelling mistakes made by children are phonological (Bloom, 2002) and audio provides feedback on the phonetics of a word while images do not. Another possible explanation is that many words that children learn are concrete in that they denote physical objects (e.g., 'bird' or 'ball') and will likely have images that represent them well, whereas more abstract concepts ('democracy' or 'because') do not.

The findings presented show that augmenting the spelling interface to offer the assistance of cues in any of the conditions (audio only, images only, or both audio and images)results in a statistically significant improvement over having no cues at all. The audio only condition showed the best results in terms of both having accurate clicks and avoiding a pattern of resorting of the first available option.

6.2. Participatory design

In our study of multimodal cues presented in Section 6.1 we observed that not all children noticed or utilized the misspellings or suggestions without being prompted. In order to create potential solutions to these issues, we used participatory design where child participants were design partners using the Cooperative Inquiry method (Fails et al., 2013; Guha et al., 2013). This process helps us understand their needs and how to ensure we meet

[5] Amazon Polly: https://aws.amazon.com/polly/.

Table 9
Analysis of spelling suggestions. *Clicks* are the number of suggestions clicked at position K; *Correct* is the number of clicks that correctly matched the child's intent and % is the proportion of clicks that matched their intent.

K	No cues			Audio only			Images only			Both audio & images		
	Clicks	Correct	%	Clicks	Correct	%	Clicks	Correct	%	Clicks	Correct	%
1	28	21	.75	16	16	1.0	16	12	.75	12	9	.75
2	4	2	.50	12	10	.83	13	12	.92	14	13	.93
3	4	3	.75	19	18	.95	7	5	.71	16	14	.88
4	8	5	.63	9	9	1.0	7	7	1.0	21	19	.90
5	3	1	.33	9	7	.78	8	8	1.0	9	9	1.0
Total	47	32	.68	65	60	.92	51	44	.86	72	64	.89

them. The ten child participants involved in our design sessions were 6–11 years of age and are members of an inter-generational design team that meets twice a week. The children on the design team who were involved in the design sessions were separate from those in Section 6.1. 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.

During four participatory design sessions over a period of a month, ten child participants worked along with 4–5 adults made up of the authors and other student researchers. Each of the design sessions took 30–60 min and was completed online using teleconferencing software. Together they worked to design a spellchecking interface with a focus on bringing attention in two facets. The first being on how to better indicate that a word is misspelled; and, the second being on how to improve the interface to get users to click/interact with the misspelled word and select one of the suggestions.

Children interacted with a basic spellchecking interface which only marked spelling errors by coloring the text of the word red and underlining it and produced spelling suggestions when clicked or tapped on. The ten children on the team were split into groups consisting of 2–3 children and 1 adult. Each of the groups worked collaboratively to come up with ideas on how to improve the spellchecker interface without taking away from the search process. Several iterations of design sessions were completed with modifications performed on the interface in-between each of the sessions based on the feedback received. Children and adults worked collaboratively using the "big paper" design technique to draw out their ideas on paper (examples can be seen in Fig. 16) and the "sticky note" design technique to identify their individual likes, dislikes, and design ideas (examples can be seen in Fig. 17) (Fails et al., 2013).

The inter-generational design team designed several ideas to improve the interface and a still image of the interface with the final design ideas is shown in Fig. 18. Common and well-liked ideas included an animated circle around misspelled words like a teacher would on a paper, audio feedback (bell or chime sounds), an option to close the spellchecker if no suggestions were suitable, and automatically displaying spelling suggestions such that a user would not have to click on the misspelling. Children unanimously decided that using red on words was the best color to indicate that it was misspelled, they enjoyed that it "talked", and found the images for the suggestions engaging. Dislikes related to the images included were too small and that the pictures often just showed a picture of the word itself.

As children were more exposed to the chime sound that was played when a misspell occurred, they expressed their dislike of it. This lead to the idea of using synthesized speech to alert the user of the misspelling, similar to the synthesized speech used for the spelling suggestions, however they also found this to be repetitive. To assist with this, we included some variations of the original phrase to be chosen randomly that included "Did you mean one of these?", "What about these?", "Is this what you

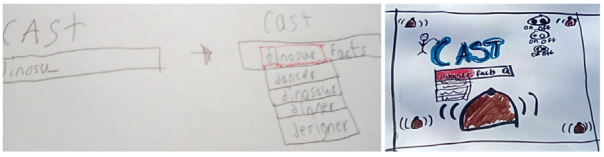


Fig. 16. Big paper examples from the first design session.

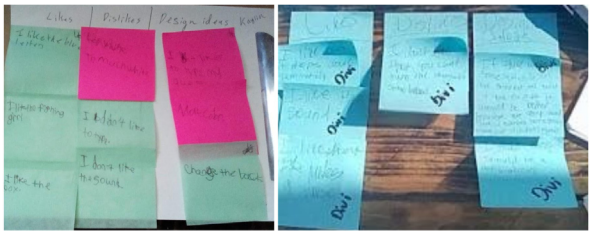


Fig. 17. Sticky note examples from the second design session.

Fig. 18. Still Image of the final interface after incorporating design ideas from children in participatory design sessions.

meant?", etc. We additionally included some pause/cue phrases (e.g., "um","hmm", or "ok") at the beginning of phrases when the voice was interrupted because of another misspelling occurring. This follows spoken dialogue research that suggests that dialogue interruptions and resumptions should start with various lexical cue phrases (Edlund, Edelstam, & Gustafson, 2014).

The idea to automatically display spelling suggestions was further modified by children's feedback to automatically go through suggestions and ream them aloud one by one when a misspelling occurred. While the word was read aloud, their associated images would be displayed next to them as well. This also produced the idea to include a speaker button next to the spelling suggestions to indicate that you could click to play the word aloud and see the picture again.

6.3. Discussion

Overall, our findings in these design sessions, which align closely with previous research with regards to children's attentiveness towards audio and visual cues, advance prior work by addressing issues previously identified (Downs, Shukla et al., 2020). We identified that children often did not click on their intended word when given a list of spelling suggestions and often preferred the options higher on the list. While this tendency was not found in the study by Figueredo (2006), this aligns with children's behavior observed in other studies on search engine result pages (Gwizdka & Bilal, 2017) and query suggestions (Anuyah et al., 2018). While audio and/or image cues are helpful in helping children make their suggestion, they each have their drawbacks. The use of images presents issues when representing words that are not concrete in that they do not denote physical objects and may require some curation to perform optimally. Audio is not always available or appropriate in the context and the synthesized speech cannot be relied upon to always pronounce words correctly. However, outcomes emerging from this study demonstrate the importance and benefits of having audio and visual cues to accompany spelling suggestions when presenting them to children.

The participatory design sessions also unveiled some problems remaining with spellchecking. In one case, a child participant commented positively on the spellcheckers ability to find their intended word and in another they noticed that their intended word was not in the list of suggestions. While many children expressed that they liked the pictures, the method for generating pictures did not always provide meaningful images as noted by one child ("the picture is just the word"). One child expressed a design idea to turn spelling correction into a game ("you could guess a letter and it would tell if you were right or wrong"). Such a system could be more effective at teaching children how to spell while correcting their spelling. While not implemented, more unique ideas came out of this as well that included haptic feedback (via a "vibrating keyboard"), "notifications" like you might see on a mobile device, and increasingly pronounced indicators of the misspelled words until they were fixed. We leave for future work formal evaluations investigating the impact of these changes.

7. Conclusion

In this study, we presented our research advances and solutions to spelling correction for children and the design of an effective spellchecking interface for children. Based on the knowledge of children's spelling habits, we theorized the use of a phonetic encoding to find suitable spelling suggestions for children's spelling errors. This method greatly outperformed state-of-the-art methods while maintaining comparable efficiency. This method proved especially impressive when generating relatively few candidate spelling suggestions which could quickly and reliably provide the intended word making the task of ranking easier. The unique application of lambdaMART LTR for spelling suggestions outperformed other machine learning methods and significantly improved the ranking of the generated spelling candidates. KidSpellλ's improvement over state-of-the-art methods was significant regardless of the context the spelling errors were made in, the grade level, spelling level, or native language of the user. Analysis of the relative feature importance from these models reinforces the importance of the KidSpell phonetic algorithm and shows the importance of other features when addressing a child audience such as age of acquisition and word frequency.

We further documented problems and addressed them at the user interface level. Modalities (i.e., audio and visual cues) were identified to assist children in their selection of spelling suggestions. The study of those modalities for a spellchecking interface demonstrates their importance on improving children's effective use of a spellchecker. Conditions using the audio only cues showed the most promise. Those results led to documented gaps and the design of an interface involving children as design partners. This gave us insights on how to improve spellchecking interfaces going forward that encourage children to address their spelling errors.

The advancements made in this study on child spelling correction could have benefits in making search more accessible for children and could improve children's experience and success while performing online searches using both commercial search engines and those that natively include KidSpellλ, such as CAST.[6] While designed and evaluated in search settings, the implications could extend to other type-based applications such as word processing software. *Information Retrieval* tasks that require textual input, such as for generating query suggestions and search results, could be further supported for child audiences. The advancements made could also benefit other programs or tasks where typing is a necessary interaction. The findings on spelling correction as well as the contribution of a child-made spelling error dataset could benefit other *Natural Language Processing* tasks where a child-written text is involved. We make the spelling correction data and algorithms used publicly available.[7]

7.1. Limitations and future work

While we made great progress on spelling errors made from both hand-written essays and typed search queries, we did not have as much success on typed spelling errors. Our focus was on improving search and typed search queries proved to be the most difficult to correct. In this instance, hand-written essay errors are not a perfect proxy for the type of errors children can make while making search queries. More training data (including query misspellings) could help with improving the overall ranking and handling of typing errors. Additionally, incorporating boundary error detection could resolve some errors more commonly made while typing. Since the focus of this work is on English-language spelling using a Roman-alphabet character set which may impact how users with a different native language may do things differently.

Further work related to spelling error detection remains. For example, some spelling errors in our dataset are meant to be pop-culture terms (e.g., *optimus prime* or *roblox*). Since these words are not in our static dictionary, it is not possible to correct them with our current method. Additional examination on the effect of image cues could be explored that limits images to those that have concrete connections to words or curating the images that are used. When it comes to a good spellchecking interface, work remains to investigate how we can best teach children how to spell rather than simply fix errors.

8. Selection and participation

The observation data of using the spelling suggestions on the search tool with various modalities were gathered at informal educational STEM events where children (n=191, ages 6–12) were interacting with computers, robots, math, engineering, and other STEM related items. Only observational data and interaction logs were collected, children were not surveyed regarding their use of the system. Our team had two stations one where children

[6] http://cast.boisestate.edu/.

[7] https://github.com/BSU-CAST/KidSpell.

could program robots and another where they could search for STEM-related and local-area information.

With regards to the essay data collected from the university literacy program, parental consent and child assent was obtained for anonymized data collection at the beginning of each semester. Neither is required for participation in the literacy program; a child's experience in the program does not vary depending upon whether or not consent was provided. Writing samples and related data from students and families who did not provide consent were not included in this study.

The speech synthesis preliminary study was conducted with 10 children (ages 6–11) who were part of an intergenerational design team that meets twice a week after school. Those children were recruited via publicly posted flyers and localized social media platforms. The purpose of the investigations were explained to participants and their parents. Parents signed consent forms to allow their children to participate, and children assented to participate. Child design partners receive a technology gift at the end of the year (valued at up to $120 USD).

Both the intergenerational design team and STEM event observation and logging protocols were approved by the institutional review board.

Acknowledgments

This work is supported by United State of America National Science Foundation Award #1763649. The authors thank the children who participated as members of the intergenerational design team (Kidsteam) and their parents for supporting them. Thanks to the children (and their parents) who participated in the other user studies. Thanks also to Aprajita Shukla, Mikey Krentz, Oghenemaro Anuyah, Teagan Mackey, Dhanush kumar Ratakonda, and Garrett Allen for their assistance preparing data and conducting studies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Machine learning evaluations

As described in Section 4.2, we considered a number of machine learning models for leveraging proposed features and identifying more suitable spelling suggestions. Inspired but the research works reported in Fomin and Bondarenko (2018), Huang et al. (2013), we considered logistic regression, a decision tree, random forest, and a multilayer perceptron (MLP), as well as Learning-to-rank models.

We conducted a number of empirical evaluations on the aforementioned models, which we trained and evaluated using the CHILDRENS$_{MSP}$ dataset described in Section 3. Fig. A.19 and Fig. A.20 capture the hit-rate and MRR for the machine learning models explored. These results further emphasize our decision to use *lambdaMART* for KidSpellλ (as described in Section 4.2).

Each model was optimized to use the following hyperparameters (unspecified parameters are the default values according to the scikit-learn library version 0.23.2):

Decision Tree (DecTree):

Max Depth: *5*
Criterion Function: *entropy*

Random Forest (RandForest):

Max Depth: *5*
Number of Estimators: *50*
Criterion Function: *entropy*

Multilayer Perceptron (MLP):

Hidden Layer Sizes: *150,100,50*
Activation Function: *tanh*

Logistic Regression (LogReg):

Regularization: *L2*
C: *10*
Solver: *lbfgs*

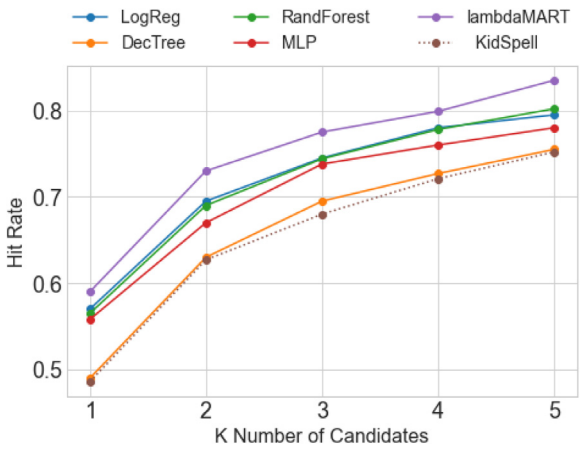


Fig. A.19. Hit-Rate for various k (number of suggestions) on spelling errors made in typed search queries with a comparison between different machine learning models.

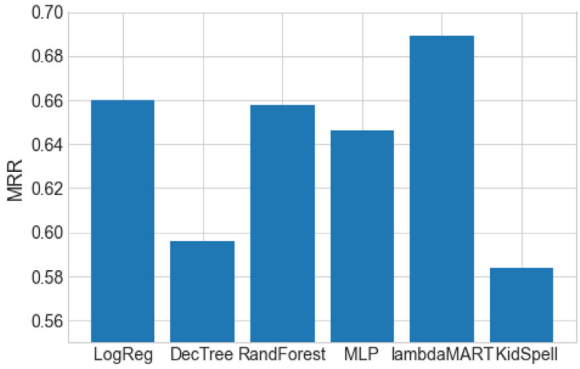


Fig. A.20. MRR using top 5 suggestions for typed search queries with a comparison between different machine learning models.

References

Anuyah, O., Fails, J. A., & Pera, M. S. Investigating query formulation assistance for children. In *Proceedings of the 17th ACM conference on interaction design and children* (pp. 581–586).

Aspell, G. (2020). GNU Aspell 0.60.8. URL http://aspell.net. Accessed 2020.

Azpiazu, I. M., Dragovic, N., Pera, M. S., & Fails, J. A. (2017). Online searching and learning: YUM and other search tools for children and teachers. *Information Retrieval Journal*, *20*(5), 524–545.

Bear, D. R., Invernizzi, M., Johnston, F., & Templeton, S. (1996). *Words their way: Word study for phonics, vocabulary, and spelling*. Merrill.

Bilal, D., & Boehm, M. (2017). Towards new methodologies for assessing relevance of information retrieval from web search engines on children's queries. *Qualitative and Quantitative Methods in Libraries*, *2*(1), 93–100.

Bloom, P. (2002). *How children learn the meanings of words*. MIT press.

Brill, E., & 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* (pp. 286–293). Association for Computational Linguistics.

Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. *Learning*, *11*(23–581), 81.

Chen, Q., Li, M., & Zhou, M. (2007). Improving query spelling correction using web search results. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CONLL)* (pp. 181–189). Prague, Czech Republic: Association for Computational Linguistics, URL: https://aclanthology.org/D07-1019.

Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search engines: Information retrieval in practice, Vol. 520*. Addison-Wesley Reading.

De Amorim, R. C., & Zampieri, M. (2013). Effective spell checking methods using clustering algorithms. In *Proceedings of the international conference recent advances in natural language processing RANLP 2013* (pp. 172–178).

Deorowicz, S., & Ciura, M. G. (2005). Correcting spelling errors by modelling their causes. *International Journal of Applied Mathematics and Computer Science*, *15*, 275–285.

Downs, B., Anuyah, O., Shukla, A., Fails, J. A., Pera, S., Wright, K., et al. (2020). KidSpell: A child-oriented, rule-based, phonetic spellchecker. In *Proceedings of the twelfth international conference on language resources and evaluation*. LREC 2020, European Language Resources Association (ELRA), https://aclanthology.org/2020.lrec-1.857/.

Downs, B., French, T., Wright, K. L., Pera, M. S., Kennington, C., & Fails, J. A. (2019). Searching for spellcheckers: What kids want, what kids need. In *Proceedings of the 18th ACM international conference on interaction design and children* (pp. 568–573).

Downs, B., Shukla, A., Krentz, M., Pera, M. S., Wright, K. L., Kennington, C., et al. (2020). Guiding the selection of child spellchecker suggestions using audio and visual cues. In *Proceedings of the interaction design and children conference* (pp. 398–408).

Dragovic, N., Madrazo Azpiazu, I., & Pera, M. S. (2016). "Is sven seven?" A search intent module for children. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 885–888).

Druin, A., Foss, E., Hutchinson, H., Golub, E., & Hatley, L. (2010). Children's roles using keyword search interfaces at home. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 413–422).

Duarte Torres, S., Hiemstra, D., & Huibers, T. (2013). Vertical selection in the information domain of children. In *Proceedings of the 13th ACM/IEEE-CS joint conference on digital libraries* (pp. 57–66).

Edlund, J., Edelstam, F., & Gustafson, J. (2014). Human pause and resume behaviours for unobtrusive humanlike in-car spoken dialogue systems. In *Proceedings of the EACL 2014 workshop on dialogue in motion* (pp. 73–77).

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., & 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* (pp. 109–120).

Figueredo, L. (2006). *Children's use of the spell checker during the composing process* (Ph.D. thesis), University of Alberta.

Fomin, V., & Bondarenko, I. Y. (2018). A study of machine learning algorithms applied to GIS queries spelling correction. *Komp'Juternaja Lingvistika I Intellektual'Nye Tehnologii*, *2018*(17), 185–199.

Gadd, T. (1990). PHONIX: The algorithm. *Program*.

Ganjisaffar, Y., Zilio, A., Javanmardi, S., Cetindil, I., Sikka, M., Katumalla, S., et al. (2011). Qspell: Spelling correction of web search queries using ranking models and iterative correction. In *Spelling alteration for web search workshop* (p. 15).

Gentry, J. R. (2000). A retrospective on invented spelling and a look forward. *The Reading Teacher*, *54*(3), 318–332.

Goodman, J. T. (2001). A bit of progress in language modeling. *Computer Speech and Language*, *15*(4), 403–434. http://dx.doi.org/10.1006/csla.2001.0174.

Gossen, T., Low, T., & Nürnberger, A. (2011). 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* (pp. 1115–1116).

Gossen, T., Nitsche, M., & Nürnberger, A. (2012). Knowledge journey: A web search interface for young users. In *Proceedings of the symposium on human-computer interaction and information retrieval* (pp. 1–10).

Greenberg, D., Ehri, L. C., & 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., & 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.

Gwizdka, J., & Bilal, D. (2017). Analysis of children's queries and click behavior on ranked results and their thought processes in google search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval* (pp. 377–380).

Hourcade, J. P. (2015). Child-computer interaction. Self, Iowa City, Iowa.

Huang, Y., Murphey, Y. L., & Ge, Y. (2013). Automotive diagnosis typo correction using domain knowledge and machine learning. In *2013 IEEE symposium on computational intelligence and data mining (CIDM)* (pp. 267–274). IEEE.

Joshi, R. M., Treiman, R., Carreker, S., & Moats, L. C. (2008). How words cast their spell. *American Educator*, *32*(4), 6–16.

Kuhn, A., Cahill, C., Quintana, C., & Schmoll, S. (2011). Using tags to encourage reflection and annotation on data during nomadic inquiry. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 667–670).

Kuperman, V., Stadthagen-Gonzalez, H., & Brysbaert, M. (2012). Age-of-acquisition ratings for 30,000 English words. *Behavior Research Methods*, *44*(4), 978–990.

Landoni, M., Matteri, D., Murgia, E., Huibers, T., & Pera, M. S. (2019). Sonny, cerca! evaluating the impact of using a vocal assistant to search at school. In *International conference of the cross-language evaluation forum for european languages* (pp. 101–113). Springer.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10*, 707–710.

Li, Y., Duan, H., & 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* (pp. 10–14). Citeseer.

Michaelis, J. E., & Mutlu, B. (2019). Supporting interest in science learning with a social robot. In *Proceedings of the 18th ACM international conference on interaction design and children* (pp. 71–82).

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

Nesset, V., & Large, A. (2004). Children in the information technology design process: A review of theories and their applications. *Library & Information Science Research*, *26*(2), 140–161. http://dx.doi.org/10.1016/j.lisr.2003.12.002.

Pande, H. (2017). Effective search space reduction for spell correction using character neural embeddings. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: volume 2, short papers* (pp. 170–174).

Philips, L. (1990). Hanging on the metaphone. *Computer Language*, *7*(12), 39–43.

Sadoski, M., Goetz, E. T., & Fritz, J. B. (1993). A causal model of sentence recall: Effects of familiarity, concreteness, comprehensibility, and interestingness. *Journal of Reading Behavior*, *25*(1), 5–16.

Santos, R. L., Macdonald, C., & Ounis, I. (2013). Learning to rank query suggestions for adhoc and diversity search. *Information Retrieval*, *16*(4), 429–451.

Sluis, R., Weevers, I., Van Schijndel, C., Kolos-Mazuryk, L., Fitrianie, S., & Martens, J. (2004). Read-It: five-to-seven-year-old children learn to read in a tabletop environment. In, *Proceedings of the 2004 conference on interaction design and children: building a community* (pp. 73–80).

Uro, G., & Lai, D. (2019). *English language learners in America's great city schools: demographics, achievement, and staffing*. Council of the Great City Schools, URL: https://eric.ed.gov/?id=ED597915.

Wang, C., & Zhao, R. (2019). Multi-candidate ranking algorithm based spell correction. In *ECOM@ SIGIR* (pp. 1–9).