Typing Race Games as a Method to Create Spelling Error Corpora

Paul Rodrigues, C. Anton Rytting
University of Maryland Center for Advanced Study of Language (CASL)
7005 52nd Avenue
College Park, MD 20742
E-mail: prr@umd.edu

Abstract

This paper presents a method to elicit spelling error corpora using an online typing race game. After being tested for their native language, English-native participants were instructed to retype stimuli as quickly and as accurately as they could. The participants were informed that the system was keeping a score based on accuracy and speed, and that a high score would result in a position on a public scoreboard. Words were presented on the screen one at a time from a queue, and the queue was advanced by pressing the ENTER key following the stimulus. Responses were recorded and compared to the original stimuli. Responses that differed from the stimuli were considered a typographical or spelling error, and added to an error corpus. Collecting a corpus using a game offers several unique benefits. 1) A game attracts engaged participants, quickly. 2) The web-based delivery reduces the cost and decreases the time and effort of collecting the corpus. 3) Participants have fun. Spelling error corpora have been difficult and expensive to obtain for many languages and this research was performed to fill this gap. In order to evaluate the methodology, we compare our game data against three existing spelling error corpora for English.

Keywords: spelling errors, corpora, games

1. Introduction

This paper presents a new method for the creation of spelling error corpora designed to be quick and cheap for the administrator and fun for the participant. The method involves the administration of a typing game in which participants race against the clock as well as each other, competing for a high score that is based on speed and accuracy. Our primary interest is the collection of errors that would occur in informal and unedited user-generated content for the purpose of improving electronic dictionary lookup systems.

Spelling error corpora, consisting of natural spelling errors paired with their correct spelling, are used to train error models in supervised noisy-channel spell correction algorithms, such as Kernigan et al (1990), Church & Gale (1991), Brill & Moore (2000), Toutanova & Moore (2002), and Boyd (2009). Spell correction algorithms such as these have been applied to very few languages, however, as there are few spelling error corpora large enough to train such systems. For English, there are several options available, but for other languages, a researcher is lucky to find one.

Spelling error corpora could be collected using spelling tests or by combing through essays. A different approach is to automatically analyze natural text for errors, and mine for the correct answer (Whitelaw et al., 2009). Ahmad & Kondrak (2005) monitored search queries and the corrections searchers made to the queries if their initial search query failed. Another approach is to mine edits on wikis (e.g., Wikipedia) for corrections of spelling errors. For example, Max and Wisniewski (2010) mine Wikipedia edits to find spelling errors in French. Priva (2010) introduced a method to collect a typing corpus over the web, but was interested in key timing information for psychological effects, not spelling errors.

Naturally, there are different types of spelling errors, including typographic errors, errors relating to phonology (e.g., selection of a homophone inappropriate to the context), and errors due to forgetting or not knowing an unpredictable word's proper spelling.

The various approaches to building error corpora will likely differ to some degree in the kinds of spelling errors they yield. For example, many of the minor edits in Wikipedia involve normalization of mechanics and style, such as capitalization and diacritics. The mining of search queries may yield misspellings due to genuine uncertainty, where the spelling is particularly difficult (such as infrequent technical terms) or unpredictable (such as proper names). An audio transcription task (where users type spoken language) would be expected to highlight phonological errors and mishearings.

A speed typing task is calculated to collect errors that arise from fast typing, such as typographical errors (where the user knows the correct spelling, but fails to correct the error). If it also yields errors similar to those from mining search queries, it may provide a useful complement to that technique when no search query corpus of sufficient size is available, and analogously with other types of error corpora and techniques listed above. While a corpus derived from query logs would be ideal for our comparison, we have no such corpus at our disposal, and the original query collection tool used by Ahmad & Kondrak (2005) seems to be unavailable, making exact replication of their work impossible. Therefore we have compared our results to standard, general purpose, readily available error corpora.

2. Existing Spelling Error Corpora

The Birkbeck Spelling Error Corpus (Mitton, 1985) contains a large number of corpora compiled from several different studies of English, from spelling test to natural e-mail errors. Across all the corpora in this dataset, there are 36,133 misspellings of 6,136 words. Pedler and Mitton (2010) distribute two "real-word" spelling error corpora for English, consisting of a total of 6775 words. A "real-word" spelling error is said to occur when a valid word in English gets used in an incorrect way, and where that word is misspelled only within the context that they were found. The words in this dataset may be homonyms, rhyming words, or words that have orthographic similarity.

3. Corpus Comparison

We chose to develop the game for English in order to show that the speed typing game methodology is comparable to the approach used to develop existing spelling corpora, but there are three difficulties in performing this comparison.

Much of the work in corpus similarity has been on token-based units or higher. We are interested in character differences.

While there has been some work in calculating corpus similarity (Kilgarriff, 2001; Oakes, 2008), these comparisons are performed on the content of the documents. We are trying to find the closest corpus based on the errors made to content.

Typing errors are contextual. A hand in a certain position, and moving to a certain position, has the tendency to make errors that are relative to those positions. Likewise, a phonologically-motivated spelling error may be conditioned on the sounds of the letters before and after the target character. Any evaluation should take these contextual triggers into account.

Because of these dependencies, we illustrate our errors using rules that are triggered by a left and right character context, similar to the rules found in phonology.

Sekine (1997) examined the use of corpus-extracted rules to calculate corpus similarity for genre classification. The author extracted syntactic derivation rules from multiple corpora, and calculated the cross-entropy to other corpora, both in-domain and out-of-domain. He found that the cross-entropy of corpora within a genre were smaller than the cross-entropy out-of-domain, and he was able to use these cross-entropy scores to cluster a corpus to a genre. The author was then able to use these clusters to perform model selection for syntactic parsing, thereby improving the performance of his parser.

We use the rule-based cross-entropy calculation in this paper, comparing the errors found in our elicited corpus to the errors present in existing corpora.

4. Method

4.1 Qualification Test and Location Filter

Participants were found through an advertisement for the system on Amazon Mechanical Turk (AMT). AMT is a website that allows workers to meet employers. These workers expect to perform small tasks in return for a small payment. A typical task might involve the classification of a website into genre category, and the typical payment for such a task might be a penny per classification.

Interested participants were given a qualification test designed to filter out non-native speakers of English. The participants were presented with a list of 18 radio buttons labeled "*Language Name* is my native language." in the language and script of each language. After these 18 radio buttons, a textbox was presented with "My native language is _____" translated into each of the 18 languages. If a user responded with a language other than English, their user ID would not be permitted to play this particular game. While this is a simple test, it helped to reduce the number of non-native English speakers performing the experiment.[1]

The game was available to AMT users located in United States, Canada, New Zealand, England, and Australia, and data were collected for about two weeks.

4.2 Participants

251 participants completed the experiment, receiving $0.10 for their participation. Several English keyboard layouts were presented as images on the screen, and subjects were asked which one most closely matched their keyboard. 98.91% reported that they used a QWERTY keyboard, 2 reported that they used a Dvorak, and one reported that they used a keyboard other than those two. 93.23% accessed the experiment from the United States, 4.78% from Canada, 1.59% from United Kingdom, and 0.40% from Australia. 84.46% self-reported that they were right-handed, 10.36% reported left-handedness, and 5.18% reported that they were ambidextrous. 75.30% reported more than 10 years of computing experience, 19.92% had 5-10 years, 4.38% had 2-5 years, and 0.4% had 2-5 years experience.

To reduce the amount of variables in the data, and to allow our results to be comparable with other publications in the typing community, all non-Right Handed, non-QWERTY participants, who reported that they looked at the keyboard typically, were removed from the results demonstrated in the following sections. In order to compare our results to existing corpora, we chose to only include typing results from the United States[2]. 153 participants remained.

4.3 Stimuli

Two sets of wordlists from the Birkbeck Spelling Error Corpus (Mitton, 1985), Masters (1927) and Angell et al.

[1] One reviewer suggested delivering a test of English usage to filter out non-native speakers. It would be difficult to create such a test that would discriminate between the two categories reliably. Additionally, it would pose language knowledge constraints on the extension of this system to languages the researchers are less familiar with.

[2] Additionally, 8 subjects were found to have more or less than the expected number of responses. The data for these participants are not part of the remaining calculations.

(1983), were combined into one stimulus list. Masters (1927) contains lists of words that were presented in a spelling test task to 8th-graders, high-school seniors, and college seniors in the United States. All words were misspelled by at least one college senior. All words on the list were included in our stimuli. Additionally, words were used from Sheffield (Angell et al., 1983), a collection of words with spelling and typographical errors collected from an academic department at Sheffield University.

Words were removed that were under a Google hit count of 1,000,000, that were British or Canadian variants, or were expected to be unfamiliar to the participants (e.g. mustn't, diagrammatically, despatched, uncritical, internecine). The Masters and Sheffield databases, combined with the filtering steps outlined here, result in a list of 562 words of American English spelling verified to be current.

4.4 The Game

Mechanical Turk users entered the game and were asked questions about their demographics and keyboarding history. Users were then presented with 100 virtual screens, with one word presented on each screen. Users were asked to type the word presented to them underneath the prompt, and were asked to press the *ENTER* key after each word. The *ENTER* key automatically advanced the user to the next word. All words were loaded on the user's web browser in advance via a single-HTML file that advanced through the word list with an AJAX presentation. This was done in order to minimize the time required to advance screens. A timer at the top of the screen calculated the time that passed since the beginning the experiment. An internal key logger recorded the start time of each character press, and all keystrokes made by the user.

At the end of the experiment, users were presented with a scoreboard, and were offered the ability to post-report whether they had looked at the keyboard during the experiment.

5. Error Analysis

We collected 15,300 non-blank responses, of which 4.65% had errors. The response of each errorful word was compared to the stimuli using an implementation of edit distance that reported the minimal string edit path that would convert the stimuli into the response using the standard character insert, delete, and substitute operators, and where each operation had a cost of 1. For example, if the word *cat* were to be typed as *kat*, the minimum string edit path would be *[SUB[c,k], OK, OK]*.

Of the errors found in the corpus, 31.97% were insertions, 48.90% were deletions, and 19.12% were substitutions.

Errors are reported as rules in a trigram context where the target character is at the center of the string, and word boundaries count as characters. In the cat→kat example, the substitution operator would produce the rule

c→k/#_a

We present an overview of the most common insertion errors in Table 1, most common deletions in Table 2, and most common substitutions in Table 3. Each table contains the count of the error, the rule that describes the error, and examples from our AMT corpus.

The left side of the slash in the character rule is the derivation, and the right side of the slash is the context in which the derivation may operate. The null set character (Ø) is used to represent an insertion when it occurs on the left side of the arrow, or a deletion if it occurs on the right side. An underscore is used to represent the position of the target character. The pound character (#) is used to represent the beginning of a word when it occurs on the left side of an underscore, and the end of the word, if it is on the right side.

Cnt	Error	Examples
4	Ø →u/o_g	catalogs→catalouges, cataloguing→catalouguing
5	Ø→u/o_s	curiosity→curiousity
5	Ø→o/p_r	approach→apporach, inappropriate→inapporopriate
5	Ø→i/c_e	conceive→concieve, magnificent→magnificient
19	Ø→e/g_m	acknowledgment→ acknowledgement, judgement→judgment

Table 1: Most frequent insertion errors.

Cnt	Error	Examples
6	o→Ø/l_g	apologies→apolgies, philology→ philgy
8	n→Ø/e_t	conscientious→conscietious, conveniently→ coventiety
9	i→Ø/t_o	cancellation→cancellaiton, functional→funcitonal
9	d→Ø/e_#	accrued→accruse, administered→dministere
14	m→Ø/m_o	accommodate→accomodate, accommodation→accomodation,

Table 2: Most frequent deletion errors.

Cnt	Error	Examples
3	a→o/c_n	canceled→conceled, canceling→conceling
3	y→t/l_#	immensely→immenselt, obviously→obviouslt
3	m→n/u_#	memorandum→memorandun, minimum→minimun
4	y→e/l_#	inevitably→inevitable
5	d→s/e_#	accustomed→accustomes, associated→associates

Table 3: Most frequent substitution errors.

6. Comparing Spelling Corpora

We compare the errors made in our study against errors collected in three of the normal adult spelling corpora distributed in the Birkbeck Spelling Error Corpus.

Fawthrop ("American Typewritten") is a corpus of 809 spelling errors gathered from four studies on American spelling errors by various researchers. Sheffield ("British Typewritten") is a corpus of 384 misspellings collected from affiliates of a British university. Wing ("British Handwritten") is a collection of errors made by British high school students on a college entrance exam. For British Handwritten, we only utilize the spelling error section of the corpus. Since this is a typewritten experiment, and our data has been constrained to the United States, we would expect our corpus to have a similar distribution of errors to American Typewritten.

For each of these three test corpora, we calculate the minimal edit path from the stimuli to the response, and form trigram-context rules as we did for our AMT corpus in Section 6.

To measure the similarity of our corpus to these evaluation corpora, and to determine if the game elicits errors similar to traditionally collected error corpora, we use cross-entropy calculated on the edit operations derived from the corpora. We define 5 classes of edit operations. ALLOPERATIONS examines all edit paths, whether there was an insertion, deletion, substitution, or if the character stayed the same within the trigram context (e.g. the two "OK" letters in our cat→kat example). ALLERRORS examines only the errors-insertion deletions and substitutions together. INSERTIONS, DELETIONS, and SUBSTITUTIONS segments the rules into individual edit operations.

Sekine (1997) utilized two constraints in his system. Because infrequent rules are not representative of the corpus, Sekine discarded any rule that occurred less than 5 times. Additionally, because any rule that was representative of the corpus should occur more frequently than in other corpora, Sekine discarded any rule whose probability was 5 times less in the test corpus than the overall probability across all corpora. We utilize these constraints as well.

For each class of rules we examine all rules of that type in the AMT corpus. We calculate the probability that the particular rule occurred out of all the rules of that class that did occur. This is multiplied by the log of the probability of occurrence in the evaluation corpus. The calculation for each rule is then summed, and negated. The resulting number is the cross-entropy. The lower the cross-entropy, the closer the fit between our corpus and the evaluation corpus.

First, we perform cross-entropy on the results of all edit paths, The results of ALLOPERATIONS can be found in Table 4. We find that the American Typewritten corpus has the closest fit of the three evaluation corpora, while the British Handwritten and British Typewritten are lower and tied.

Corpus	ALLOPERATIONS
American Typewritten	0.16
British Typewritten	0.17
British Handwritten	0.17

Table 4: Cross entropy of all character rules.

Next, we calculated the cross entropy of the insertion, deletion, and substitution errors together, without including the rules representing correctly typed characters. Table 5 shows that the overall distribution of errors in the AMT error corpus is relatively close to the errors in the American Typewritten corpus, and further from the two British corpora.

Corpus	ALLERRORS
American Typewritten	1.19
British Typewritten	1.23
British Handwritten	1.25

Table 5: Cross entropy of all error rules.

Tables 4 and 5 show us that the speed typing game does collect a similar distribution of character operations as traditional spelling error corpora. In order to understand if there are any edit operations that are divergent from American Typewritten, we segment the insertion, deletion, and substitution analyses into separate tables (Tables 6, 7, and 8, respectively). We find that in regards to each specific operation, American Typewritten is the best fitting error corpus.

Corpus	INSERTIONS
American Typewritten	0.25
British Typewritten	0.26
British Handwritten	0.27

Table 6: Cross entropy of insertion errors.

Corpus	DELETIONS
American Typewritten	0.55
British Typewritten	0.57
British Handwritten	0.58

Table 7: Cross entropy of deletion errors.

Corpus	SUBSTITUTIONS
American Typewritten	0.42
British Typewritten	0.43
British Handwritten	0.44

Table 8: Cross entropy of substitution errors.

As we restricted our corpus to only those participants that were accessing the game from the United States, the results shown in Tables 4-8 were expected.

Incorporation of the responses from the non-US participants may produce a more robust pan-English spelling corpus, but would make for a more difficult evaluation.

While error corpora of query logs or Wikipedia spelling corrections were not available to us at the time of writing, we anticipate from our results that a speed typing approach would correlate well with query logs, and less well with Wikipedia spelling corrections.

7. Games for Other Languages

As mentioned earlier, this system was designed with the intention to adapt to other languages. Some languages may require additional factors for experimental control.

Languages such as Arabic have a high number of keyboard layouts that could pose a problem interpreting some errors. It would be unwieldy to display on the screen all Arabic keyboards. Another approach is to record participants as they are asked to type their keys in the order they appear on the keyboard. This is time consuming, and would bore the user.

Languages such as Punjabi utilize a font that overlays Latin encoded characters in order to display the characters of their language. These languages may require information about the font being used to project the characters they see on the screen. Capturing font information through the web browser is not possible, and the user would have to be asked to supply this information.

8. Future Work

We have adapted this experiment to Modern Standard Arabic, and are currently collecting data. Native English speakers far outnumber native Arabic speakers on Amazon Mechanical Turk (AMT), and recruiting participants is slower. Our English experiment was self-contained on AMT, but we have needed to advertise the Arabic experiment and recruit from outside the network.

Several of the errors found in the corpus were created by the reversal of two characters. Using standard edit distance, this would be recorded as two edit operations. These errors could, in the future, be calculated as one edit operation by the introduction of a swap edit operator.

9. Conclusions

We introduced a method to induce a spelling error corpus by using a competitive typing game. The approach is low-cost for the researcher, is fun for the participant, and allows the subject to participate from any web browser. We evaluate the system on three existing English spelling error corpora and show that the system collects errors similar to existing methods. Out of the three corpora, we found the American Typewritten dataset to be a close fit to the collected corpus on several cross-entropy based evaluations.

While we tested the approach on English, the collection methodology is easily adapted to other languages by using templates for the screens and word lists for the stimuli.

10. Acknowledgements

This material is based upon work supported, in whole or in part, with funding from the United States Government. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the University of Maryland, College Park and/or any agency or entity of the United States Government. Nothing in this report is intended to be and shall not be treated or construed as an endorsement or recommendation by the University of Maryland, United States Government, or the authors of the product, process, or service that is the subject of this report. No one may use any information contained or based on this report in advertisements or promotional materials related to any company product, process, or service or in support of other commercial purposes.

11. References

Ahmad, F., Kondrak, G. (2005) Learning a Spelling Error Model from Search Query Logs. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. East Stroudsburg, PA: Association for Computational Linguistics, pp. 955-962.

Angell, R.C., Freund, G.E., Willett, P. (1983) Automatic spelling correction using a trigram similarity measure. *Information Processing & Management*. 19(4), 1983, pp. 255-261.

Boyd, A. (2008). Pronunciation Modeling in Spelling Correction for Writers of English as a Foreign Language. In *Proceedings of the 6th Language Resources and Evaluation.* Marrakech, Morocco.

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.

Church, K., Gale, W.A. (1991) Probability Scoring for Spelling Correction. *Statistics and Computing*. 1(2), pp. 93-103

Kernighan, M., Church, K., Gale, W (1990). A Spelling Correction Program Based on a Noisy Channel Model. In *Proceedings of the 13th conference on Computational linguistics, Volume 2*, Helsinki, Finland: Association for Computational Linguistics.

Kilgarriff, A. (2001). Comparing Corpora. International Journal of Corpus Linguistis. 6:1. pp. 97-133.

Kukich, K. (1992) Techniques for automatically correcting words in text. *Association for Computing Machinery Computing Surveys*. 24(4), pp. 377-439.

Masters, H.V. (1927) A study of spelling errors. Ph.D. Thesis. University of Iowa.

Max, A. & Wisniewski, G. (2010). Mining naturally-occurring corrections and paraphrases from Wikipedia's revision history. In LREC'10, Valletta, Malta.

Mitton, R. (1985) Birkbeck spelling error corpus. Retrieved from http://www.ota.ox.ac.uk/headers/0643.xml 2011-07-13

Oakes, M.P.: Statistical Measures for Corpus Profling. In: Proceedings of the Open University Workshop on Corpus Profiling, London, UK (October 2008)

Pedler, J., Mitton, R. (2010). A Large List of Confusion Sets for Spellchecking Assessed Against a Corpus of

Real-word Errors. In *Proceedings of the 7th Language Resources and Evaluation,*

Priva, Uriel Cohen. (2010) Constructing Typing-Time Corpora: A New Way to Answer Old Questions. Proceedings of the 32nd Annual Meeting of the Cognitive Science Society

Sekine, Satoshi. (1997) The domain dependence of parsing. Proceedings of the fifth conference on applied natural language processing. East Stroudsburg, PA: Association for Computational Linguistics, pp. 96-102.

Whitelaw, C., Hutchinson, B., Chung G.Y., Ellis, G. (2009). Using the Web for Language Independent Spellchecking and Autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Volume 2*. East Stroudsburg, PA: Association for Computational Linguistics, pp. 955-962.