Eye-tracking without the eye-tracker

Abstract

This paper details the process of creating an iPad app for the purpose of measuring eye movements during reading. The app was meant to simulate an eye-tracking experiment as closely as possible without the corresponding equipment. The aim was to replicate established results in the field of text simplification (using eye-tracking equipment) using the app. To this end, a study was designed around minimal sentence pairs, to detect fixations, regressions, and saccades. While we found that the app is capable of detecting small differences in how people process sentences, and there are some statistically significant different phrases amongst the pairs tested, most pairs showed no difference.

1 Introduction

In 2015, the U.S. Department of Education and the National Institute of Literacy conducted a study showing that 14% of adults in the U.S. cannot read beyond a "basic level" ("The U.S. Illiteracy Rate"). For the millions of adults that fall within this category, information and resources predicated on reading is difficult to access, leading to issues with daily tasks, ranging from reading the news to acquiring healthcare. A great number of these adults may be unable to read due to learning disabilities, insufficient instruction in school, or not being native English speakers. In all of these cases, it is of vital importance to these people to have available a resource to simplify reading for them, and it is in this vein that automatic text simplification projects have been undertaken.

1.1 Purpose

This paper describes a mechanism designed to conduct research in the first step of automatic text simplification: determining what it means for a text to be "hard." Existing literature in this area relies on observing readers' reactions as they read a text, such as through eye-tracking equipment (Ashby *et al.*). Although effective, eye-tracking studies require participants to come to specialized laboratories and put on cumbersome equipment, involving a lot of effort on the part of both the researchers and the participants. If a method can be found that is just as effective, but significantly less cumbersome, much of the research in this field can be simplified, and future research can be undertaken with ease.

1.2 Background

The research of interest to this paper has been conducted in the field of eye-tracking experiments to gauge text difficulty. Rather than create models to automatically simplify texts, this research area focuses on using human readers to identify patterns in the way they read, and use this data to estimate what characteristics of a text are easy or difficult. These patterns take on three forms in particular, based on how our eyes move around naturally when reading.

1. *Fixations* refer to the short durations where the eye is fixed on a particular place or word. People usually fixate on few characters, meaning a word may require more than one fixation to be understood. [66]
2. *Regressions* occur when readers return to an earlier point in the text in order to reread it, typically due to either having missed something or requiring a reread for clarification. [17]

3. *Saccades* are the small jumps that occur between fixations, and are relatively short (10-100 ms, as opposed to fixations of typically 200-250 ms). [27]

The purpose of this experiment was to show that the app being developed was capable of detecting all three of the above behaviors, which would allow this mechanism to be useful in experiments that are like the established eye-tracking technique.

Results from existing eye-tracking experiments also provide the justification for some of the particular texts tested in this experiment. In particular, the paper “Eye movements of highly skilled and average readers: differential effects of frequency and predictability” suggests that certain pairs of words produce distinctly different reactions in readers, and that these reactions can be monitored using the three behaviors listed above. They found a statistically significant difference between word pairs where one word was more “predictable” than the other, from the context of the same sentence. It is studies such as these which give us an idea of which characteristics of texts may be worth studying.

2 App

In order to carry out the study, we designed an iPad app that would allow us to test many different passages and get an idea of whether we could replicate eye-tracking results.

2.1 Functionality

In terms of features, this application was designed specifically for this study, and so its functionality is limited to precisely that which was needed for the experiment.

This app was designed with minimalism in mind. For the purposes of this experiment, it was necessary to keep track of individual data sets while preserving the participant’s anonymity. To this end, the first function the app supports is the ability to assign an ID number to each set of data readings. This ID number is filled into a text field by the researcher leading the study.

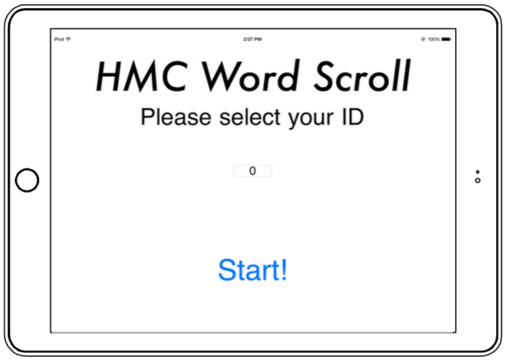

Figure 1: Home screen.

From the home screen, the participant must click “start” in order to begin the experiment. When this button is pressed, the iPad calibrates itself, meaning whatever position it is held in at that moment is treated as a zero tilt. After this, the user is presented with the reading screen, which consists of a reset button and a horizontal reading frame (a rectangular box in the center of the screen). Upon tilting, text appears inside the reading frame and scrolls through.

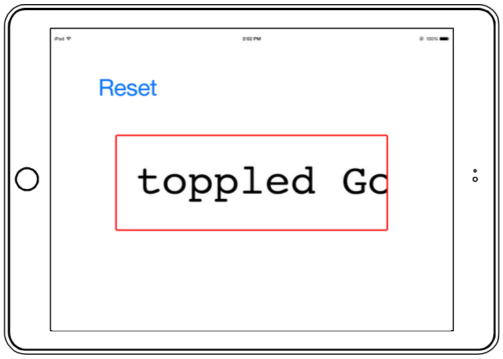

Figure 2: Reading screen.

The accelerometer measures the acceleration as the user tilts the iPad, which is used to determine how fast the text scrolls. The text scrolls from right to left, and the user can make the text scroll forward by tilting the iPad to their right, as well as backward by tilting the iPad to their left. This functionality allows the user to read at their own pace, as well as affords the user the flexibility to reread passages if they so choose.

Many of the settings behind the app were determined through a user study. In the study, participants were given different options for settings for the following parameters:

- Tilt sensitivity
- Reading frame size
- Font size
- Font choice
- Tilt direction

Thus, the above settings were uniform across participants for this text comparison study. Data is collected by the accelerometer built into the iPad, which reads the device's acceleration every 0.2 seconds, and associates this value with the time elapsed and the text visible on the screen at that time.

3 Experiment

The app was specifically designed for this specific study, which was carried out over the course of three days. Participants were recruited from summer research students, and during the course of the experiment, were asked to read 12 passages. Each passage has an "A" and "B" version.

3.1 Text Pairs

The texts used in the experiment were constructed based on sentence pair comparison. The idea behind this technique is as follows: in an effort to identify which aspects of a text are "easy" or "hard" to read, sentence pairs are constructed, where one sentence serves as control ("B") and the other as test ("A"). The two sentences differ from each other in small but significant ways. Asking multiple users to read version A, and asking multiple different users to read version B, allows us to compare whether, on average, the differences between A and B had a significant impact on the "difficulty" of that passage. Each of the 12 passages the participants read contained three such target sentences, giving us a total of 36 sentence pairs to compare.

There are three kinds of differences that were being tested using the sentence pairs. With four texts for each kind of difference, and each text containing three target sentence, there are 12 sentence pairs per type of difference. These types are as follows.

Lexical. A lexical sentence pair consists of two sentences that are identical except for one word. These words must be synonyms of one another, where one of them (the B version) is expected to be more "difficult" than the other. An example is given below:

A. "... believes that he escapes *liability*... "
B. "... believes that he escapes *responsibility*... "

Semantic. A semantic sentence pair also consists of two sentences that are identical except for one word. These words are equally correct from a grammatical point of view, but one of them (the B version) is unexpected in the context of the passage, while the other (the A version) must be highly predictable. For instance:

A. "... mailed her a *compass* from China... "
B. "... mailed her a *letter* from China... "

Syntactic. A syntactic sentence pair consists of a garden path sentence (the B version) and another sentence (the A version) which means precisely the same thing as the garden path sentence, but with a clearer syntactic structure. A garden path sentence is one where a reader begins to apply a certain syntactic structure, only to reach a point in the sentence where the structure no longer applies. The addition of some qualifiers explains the intended syntactic structure. Consider the following:

A. "The horse raced past the barn *fell*."
B. "The horse *which was* raced past the barn fell."

4 Data Analysis

4.1 Acquisition

Data analysis consisted of two stages, the first run while the app was operational, and the second post-testing. During the operational phase, the indices of the characters on screen were calculated every update step. At this time, the accelerometer reading was converted into characters per second using a linear factor which included the size of each character, the time since last update, and the size of the viewing window. (Note that the size of each character was the same by design, since we chose to use a monospaced font to make such calculations easier.) Thus, each reading yields a time stamp, indices representing the word on screen at that time, and the "reading speed" of the user at the time.

4.2 Processing

After testing was completed, this data was further analyzed by a separate program. For each text, the program first gathers all instances of words read by the user; that is, words whose indices had been recorded during the operational phase. Since these values had previously been associated with a characters per second (CPS) value, denoted x, the program then uses this data to calculate the average CPS for that particular user across all 12 texts. A second average value was calculated, this time an average "speed" for each of the texts, which we denote μ_i, where i represents the text (and ranges from 1 to $n = 12$). Finally, the same word can appear on multiple readings, and so have multiple CPS values that correspond to it. An average of all of these values is calculated, denoted $\bar{x}_{i,w}$, representing the average CPS for word w occurring in text i.

Each individual word was then scored using the Z-Score, which we denote S_w. This calculation used the word's average speed, $\bar{x}_{i,w}$, the average speed of the particular text, μ_i, and the standard deviation of the CPS values fo that text, σ_i. For instances where a word was truncated, a specific fraction had to be present for the word to be considered "in-frame." A Z-Score is a measurement of a value's proximity to the mean of the distribution: a Z-Score of 0 implies the value is the same as the mean. This calculation was given by:

$$S_w = \frac{1}{n}\sum_{i=1}^{n}\frac{\bar{x}_{i,w} - \mu_i}{\sigma_i}$$

Once all users' data had gone through this processing, each word has as many Z-Scores as users that read that particular text. These multiple Z-Scores form a distribution for that particular word, with a corresponding mean and standard deviation. The mean of this distribution provides a normalized score per word across all users. We then compared Z-Score distributions for "target" words in the A and B versions of the same text in order to determine whether there was a significant difference (see Results).

Recall that each pair of texts contains three target words. After completing the above, we considered the Z-Scores of all three words in a passage as a whole. This allows us to get a larger estimation of whether the "harder" words were, in fact, more difficult than the easier ones.

5 Results

By comparing average Z-Scores of partner texts, as described above, we see that only four text pairs showed significant differences in reading speed, with $p < 0.05$ (see Table 1). A higher Z-Score indicates a higher reading speed, thus three of the four significant pairs support the hypothesis (shown in green), that the theoretically harder passages would have a lower reading speed across their three target sentences.

	Lexical		**Semantic**		**Syntactic**	
Text Number	A	B	A	B	A	B
1	0.6306	0.3272	1.1324	0.6256	−0.2520	0.3485
2	−0.2408	0.4300	0.4229	−0.0737	−0.4772	−0.4914
3	0.1956	−0.8105	−0.1849	0.0549	−0.5468	−0.1049
4	−1.9396	−0.8171	0.1283	0.8202	0.3725	0.0114

Table 1: Z-Scores across passages.

6 Discussion

6.1 Summary

x

6.2 Future Directions

x

Acknowledgments

x

References

Huffington Post. 2013. "The U.S. Illiteracy Rate Hasn't Changed in 10 Years." Huffington Post, Web.

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503–512.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.