

Reading Edge: Reading Proficiency Through Paragraph Reconstruction

Updated: September 24, 2023

Abstract

Reading Edge is built around a simple, two-step paragraph reconstruction activity whereby a student reads a paragraph and then reconstructs the same paragraph with 15-20% of the words replaced by a drop-downs of word choices. When the paragraph is correctly reconstructed the application awards points equal to the number of corrected words and the student can continue to the next paragraph. If the student misses any word then the exercise is repeated with the same paragraph but with a different set of randomly chosen words replaced.

Reading Edge is the core activity from an edu-kiosk platform developed in 2013 in which the points were cashed-in to unlock internet access. There were two problems with the original: hardware complexity and content creation. Reading Edge 2023 solves the first problem by leaving it up to the parent or teacher to define the value of earned points, ie the incentive. The second problem is solved by integrating with Wikipedia to quickly and easily produce assignments of reliable quality from article summaries and images.

Application Description

Reading Edge is built with the latest reactive web framework, called Vue.js, and is hosted on a scalable cloud server running the latest Ubuntu Linux. User authentication is handled with Google Firebase and data is stored using a combination of Google Firestore and Google RealTimeDatabase services. The source code of the project is stored in a private GitHub repository.

The application itself is a responsive WebApp which runs in a browser, there is nothing to download or install. One benefit of this is that users are always served the latest version. Another benefit is that the same code works for desktop and mobile platforms, alike.

Reading Edge does not store text or images from Wikipedia, but rather only links to text and images on Wikipedia. The exception to this is if a user modifies an article summary text, or provides their own custom text. Images are never saved, and thus all images must be publicly available on the internet. Images outside of Wikipedia can be used so long as they have a publicly accessible url, as the url is all that is saved to the database. Assignment pages generated from Wikipedia content are still attributed as such by automatically providing a link to the original article when the page is generated.

Paragraph Reconstruction

The paragraph reconstruction exercise at the heart of Reading Edge was a product of necessity to ensure accountability from the author's own children. With a desirable incentive in place the exercise eliminates the need to constantly monitor the student. For, if anyone reconstructs a paragraph with 20% of the words replaced you can rest assured that they must have read the paragraph carefully and paid attention to details like context, grammar and punctuation, while a good incentive compels them to make the necessary effort on their own.

Recommendations for Use

Careful reading and comprehension of material seem to occur naturally with sufficient incentive, whether it be genuine interest in the material or something else like internet access or money. However, the parent or teacher can further promote positive outcomes by providing material that the student is interested in. So, if the Reading Edge is being used mainly to promote careful reading itself, as opposed to careful reading about some core subject, then the assignment generator's interface to Wikipedia enables the parent or teacher to quickly produce content tailored to individual students' interests.

As an example, consider a student who is interested in airplanes. In this case a teacher could set the number of pages to a reasonable number, say 10, and enter a few comma-separated search terms like "airplane", "boeing jets" and "commercial aircraft", hit the generate button and review the results. It might be necessary to delete one or two pages that aren't exactly relevant, but generally this process takes only a minute or two, and can occupy a student with interesting content for upwards of 30 minutes.

Typical articles in Wikipedia have many associated images, as well. The parent or teacher (instructor hereafter) can configure the number of images to retrieve, and these images can stimulate further interest and compel the student to look forward to the remaining material and take pleasure in the whole endeavour of reading. In short, the system is designed to make it easy to customize material on a per-student basis.

Teacher & Student Users

Every account has two roles: teacher and student. Upon logging in the user is placed in the teacher role. Teachers can create courses and assignments. Each course has a unique course-key which the teacher can share by whatever means they like. The system does not get involved in this transaction, however. The course-key can be sent by email or written on a piece of paper, it doesn't matter. Anyone who has the course-key can switch to student role and enroll themselves by entering the course-key in the enrollment pop-up, including the instructor.

Courses & Assignments

Courses and assignments are each created from a single button click. This creates an object from a default template which can then be developed according to taste. One of the first things one typically does is to set the title of the course or assignment, for example. From a selected course, the user can then add assignments from a template with a single click. Likewise, from a selected assignment the user can add blank pages from a template with a single click. Or, using the results of a search, the user can preview pages of the search result set and click to include desired pages in the assignment. For maximum convenience and efficiency the user can let the assignment generator automatically create a set of pages from user specified search terms. This convenience is key to minimizing the work for the instructor.

Wikipedia Interface

The Wikipedia interface is found at the top of the left sidebar. It includes an interface to generate a complete assignment in one of 20+ languages from a comma-separated list of search terms. If you are creating an assignment in Spanish, for example, then you would select Spanish as the language, set the number of pages desired, and use a comma-separated list of Spanish search terms, for best results.

Below the assignment generator is the basic search control. The user can configure the number of search results to return, the language of the returned pages, and the comma-separated list of search terms to

search with. Clicking on any search term loads the corresponding article into a preview panel from which the user can decide to add the page to the current assignment or not.

Below the search control is a special feature called "On This Day", which returns roughly 10-20 articles selected by Wikipedia about notable historical events which occurred "on this day" in history, from present to ancient times. These are returned just like normal search results and are loaded into the preview panel in the same way as regular search results. You could, for example, just make an entire assignment consisting of the articles corresponding to any given day of the year. The default date is the current date, but can be changed to any date from a popup calendar.

Multi-Language Support

Reading Edge incorporates several language features. The user selects the language first, before conducting a search or generating pages. The results are then returned in the language selected. It is not meant to take a Spanish page and change the language to English. However, Wikipedia does have translations of pages to other languages, and these alternative pages are included as clickable links in the page interface once a page has been added to an assignment. Further, if a page is in Spanish, for example, then the text-to-speech functionality will speak the text in Spanish (see below). It is also possible to change the language from the page editor interface and listen to a passage written in Spanish but spoken with a French voice, which can be amusing.

Text-To-Speech

As mentioned in the previous section (Multi-Language Support), setting a language affects both the search results and the voice used to read-aloud a given passage. The text and audio are separate systems. Currently the language choices are limited to those supported by the text-to-speech system, even though Wikipedia translates to hundreds of different languages. The text-to-speech system currently supports 20+ languages, with the most recent being Thai and Kiswahili.

Burmese is one language that is not supported by the TTS system, however one could still paste Burmese text into an empty page and just not use the tts playback. Burmese text, to continue with the example, could be obtained by using Google Translate and just copy-and-pasting into a blank page. Of course you could also set the language to French and it would read the Burmese text with a French accent. Again, amusing for a few minutes but not useful.

Benefits of Paragraph Reconstruction

Paragraph reconstruction can be used as an instructional tool to reinforce reading and writing skills. It's a versatile exercise that can be adapted to various age groups and proficiency levels, making it a valuable resource for teachers and learners alike. The benefits include:

1. **Improves Reading Comprehension:** Paragraph reconstruction requires individuals to understand the logical flow of a text and how ideas are connected. This enhances their ability to comprehend and make sense of written material.
2. **Enhances Critical Thinking:** Participants need to analyze the content, identify the main ideas, and determine the appropriate sequence. This promotes critical thinking and problem-solving skills.

3. **Strengthens Language Skills:** By working on paragraph reconstruction, individuals develop their language skills, including vocabulary, grammar, and syntax. They learn how sentences fit together to create a well-structured paragraph or essay.
4. **Encourages Attention to Detail:** Reconstructing paragraphs demands careful attention to sentence structure, punctuation, and transitions. It encourages individuals to notice and correct errors in written texts.
5. **Boosts Writing Skills:** As participants engage in rearranging sentences, they gain insight into how well-constructed paragraphs are formed. This knowledge can be applied when they write their own essays and compositions.
6. **Strengthens Memory:** Engaging in paragraph reconstruction exercises can enhance memory, as it requires individuals to remember sentence details and the sequence of information.
7. **Prepares for Standardized Tests:** Many standardized tests, including language proficiency exams like TOEFL or IELTS, include paragraph reconstruction tasks. Practicing this skill can be valuable for test preparation.
8. **Cultivates Active Reading:** Paragraph reconstruction encourages individuals to actively engage with the text. Instead of passively reading, they must analyze the content and its structure.
9. **Promotes Collaborative Learning:** Group activities involving paragraph reconstruction can foster teamwork and discussion as participants work together to identify the correct sequence.
10. **Builds Confidence:** Successfully reconstructing passages boosts a learner's confidence in their reading and language skills. It's a tangible achievement that demonstrates their ability to understand and manipulate written texts.

Further Benefits of Reading Edge

While Reading Edge lends itself naturally to improvement of students' reading and comprehension skill, its flexibility with respect to content means that it can be used for teaching about much more than just reading skills, while still developing those reading skills in parallel. This bigger bang-for-the-buck was indeed the key consideration when deciding whether to develop the project to its current state.

Depending on the content of the paragraphs being reconstructed, students may gain exposure to different cultures, perspectives, and ideas, which can broaden their worldview and cultural awareness. Likewise it can be used to expose them to career options, current events, history, engineering topics and anything else that can be expressed and documented with words.

The language capabilities are another big bang-for-the-buck, making Reading Edge useful for teaching all of the above in students' native languages, as well as for foreign language teaching itself. Many combinations of these capabilities remain to be fully explored.

Guiding Principles

People have high expectations for education software, and software in general. To meet these expectations the future of Reading Edge will be guided by the goals of being sustainable, enjoyable, user-friendly, accessible, affordable, effective, convenient, efficient, reliable, versatile, customizable, engaging, adaptable, scalable, responsive and inclusive.

First Goals

In order to realize its potential two things need to be addressed first: user experience and project sustainability.

Enjoyable Experience

Although the software currently exists and is functional, much work actually remains. Ultimately the software needs to improve to the point where it meets peoples' expectations, people want to use it and people enjoy using it. The first areas in need of attention are the teacher and student dashboards.

Self-Sustaining

Despite its simplicity and advanced state of development, the success of the project will require further development and consistent, sustained attention. For this the project may seek to establish itself as a 501-c3 non-profit organization. This would enable the project to solicit funds to support further development and provide that sustained attention.

The project intends to maintain a free-tier for home use as well as a paid subscription tier for classrooms and schools. The development of this infrastructure is currently incomplete.

Future Goals

While keeping the software simple is key to its sustainability and scalability, data analytics is one area where the project intends to undergo significant future development, as this is something for which computer software is particularly well suited and capable, and is an area in which the author has considerable experience, as well.

Student Assessment

A simple gradebook is the only assessment tool that is currently implemented, but more can be done, starting with additional views to show how many attempts, and how much time it took a student to complete each page. This would enable the instructor to identify and quantify which students have the most trouble.

Keeping the application simple will limit the data available for scrutiny, but there are still many ways to combine that limited information to produce meaningful insights. Other measurable quantities are: how much time, overall, has a student spent on the system, how much time has been spent per page as a function of replacement percentage, or an instructor-defined difficulty index.

Quantify Effectiveness

When promoting Reading Edge it would be useful to be able to quantify its effectiveness. The US Department of Education operates a program called the "What Works Clearinghouse" (WWC) which performs independent assessments of education software according to established benchmarks and metrics. There is currently a surprising lack of research about paragraph reconstruction from the WWC. Existing case studies refer to group activities where students recreate paragraphs using whole sentences at a time, or merely just rewrite what they understood and try to capture the gist of the original passage. These activities were found to be effective, but they required substantial teacher involvement, and an

application such as Reading Edge can help make paragraph reconstruction a more accessible tool for educators.

As Resesarch Tool

Similar to a teacher's need to quantify student performance, the Education Institution, itself, needs to quantify performance on a national scale across schools, teachers and states. Every university with an Education department is involved in some form of research to this end, and with appropriate modifications Reading Edge can be a valuable tool for these researchers, as well. The author intends to establish the effectiveness of Reading Edge through collaboration with the What Works Clearinghouse, first, and then take those results to education researchers to seek research collaborations.

Keys to Success

Nevada and Beyond

Reading Edge is developed locally here in Las Vegas. As such, it would be particularly satisfying to see it have a positive impact at home in Nevada. Thus, while the project has potential for kids everywhere, the project would maintain a special focus on Nevada. Thus, in parallel with the effort to obtain the official blessings of the Department of Education and the like, the project intends to maintain a special focus on Nevada.

The order of operations here would be to first polish the application a bit more in order to meet peoples' expectations, and then to promote it locally throughout Nevada using a variety of platforms and channels. These would include targeted promotional advertising online through Google Ads and Social Media, as well as paper brochours, emails and other materials to libraries, schools and groups concerned with education here in Nevada.

The challenge and the goal would be to successfully engage with three distinct groups: parents, schools and homeschoolers.

Project Status

Reading Edge Organization

The author has investigated how to grow the project from its current state into a self-sustaining platform and organization supported by a small staff. Envisioned roles correspond to organizational finances, software development, product support, outreach and promotion, Social Media presence and solicitation of ongoing financial support. The author has a list of capable individuals for these roles who are ready, willing and interested as needs arise. A decision whether to pursue 501-c3 status has not yet been made.

Software Development

As it is a software project, software development will always be of primary importance. To this end the platform needs to meet users' expectations for ease-of-use, functionality and effectiveness. Powerful dashboards are at the top of the software development wish-list, followed by increased interface reactivity, a mobile app, and tools to visualize student performance and facilitate assessment.

In parallel with user-facing advancements there is an equally important need to develop project administrative tools. This includes custom tools for user management, subscription management, promotions management, website traffic analysis and ROI analysis with respect to marketing efforts.

Appendix

A1. Project History

Reading Edge was a product of necessity. It is the product that I, as a parent, would have gladly paid \$5 each day for. Around 2005 I was simultaneously pursuing a Ph.D. in physics, working full-time as a software engineer and was the parent of two young children attending public school. My kids would get home 2 full hours before me, and I wanted them to use some of that time to improve their own skills in preparation for their futures which lay ahead. Typically I would email links to various news articles which I thought they should read.

One day I caught a glimpse of my daughter's email account and was shocked to see that all of my emails remained unopened! Furthermore, they were going straight to the internet to play online games. So I developed the paragraph reconstruction activity to ensure that they read each article carefully, and a special credit-meter system that worked like a self-serve assignment kiosk where they would earn credits for each completed reading assignment which could then be used with the credit-meter to "purchase" their internet access.

This system had a profound effect on my children and our family as a whole. First, the articles which I compelled them to read taught them a lot about the world. Additional benefits included accountability, time management and self-discipline. The other major benefit was that I was not required to be the "bad guy" who constantly nagged them to read anymore. That distinction was transferred to the automated system.

By the time I returned home each night my kids would be full of interesting things to talk about and would frequently, literally beg for more work. That, to me, was perhaps the Holy Grail of education, ie having a student beg for more work.

Seeing this led me to develop the software and credit-meter system further, with the intent to share it with a wider audience. My philosophy was that "as long as I had my day job and career I could afford to give it away at cost" (the wifi credit-meter kiosk cost about \$100 for the hardware).

I found that people had a lot of interest in the project. I presented at a couple trade shows, presented at Py-Con, won a shark-tank-like competition and had collaborations with several developers and organizations.

Several factors ultimately precluded my success. These were: immature technology, level-of-difficulty both in cutting and pasting assignments together, and technical requirements to setup the credit-meter hardware. Another factor was my stubborn attempts to create a complimentary community of both parents and developers.

In 2023 I found myself with a bit of spare time, during which I discovered Wikipedia's JavaScript interface. That enabled me to "solve" the content creation problem of the core activity, namely having to cut-and-paste together reading assignments and imagery, and all issues relating to content attribution. As for the

original incentive that the credit-meter kiosk system provided, I decided to leave that as an external dependency and let people devise their own incentives.

The original credit-meter kiosk was named the "Net Dispenser", and the project was titled CREATE, which was an acronym for "Coalition for the Realization of Education Applications That Empower". All of this is still a possible future direction for the project, as well.

A2. Asymptopia Software

In 1999 I began to develop education software for my own kids. After some initial successes it became clear that with a little polishing these applications could be useful to a much larger, international audience, and thus I began to share them as opensource projects via my [website](http://www.asymptopia.com), Asymptopia Software (www.asymptopia.com).

A3. Author

The author, Charlie Cosse, earned his Ph.D. in physics in 2004 and has worked on a multitude of projects since, including international physics collaborations and experiments, a year as a Peace Corps Response volunteer developing software for the government of Guyana, startups in Real Estate and Food Delivery, and his current position developing flight simulator software. To this day, Reading Edge remains the best thing he's developed due to its potential for teaching kids about the world and, simultaneously, how to read carefully.