Release Plan - WikiTrust 2.0 - WT Team

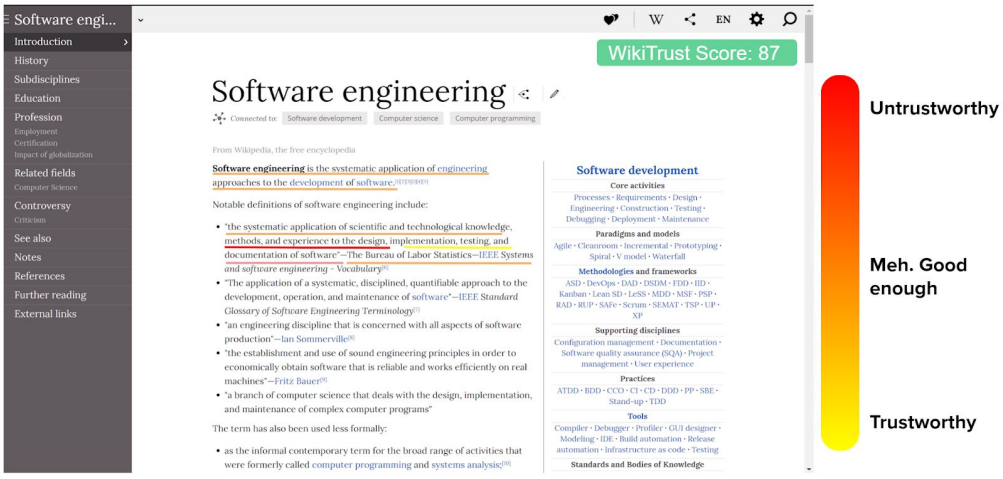
Cantaloupe Release - Dec 2019

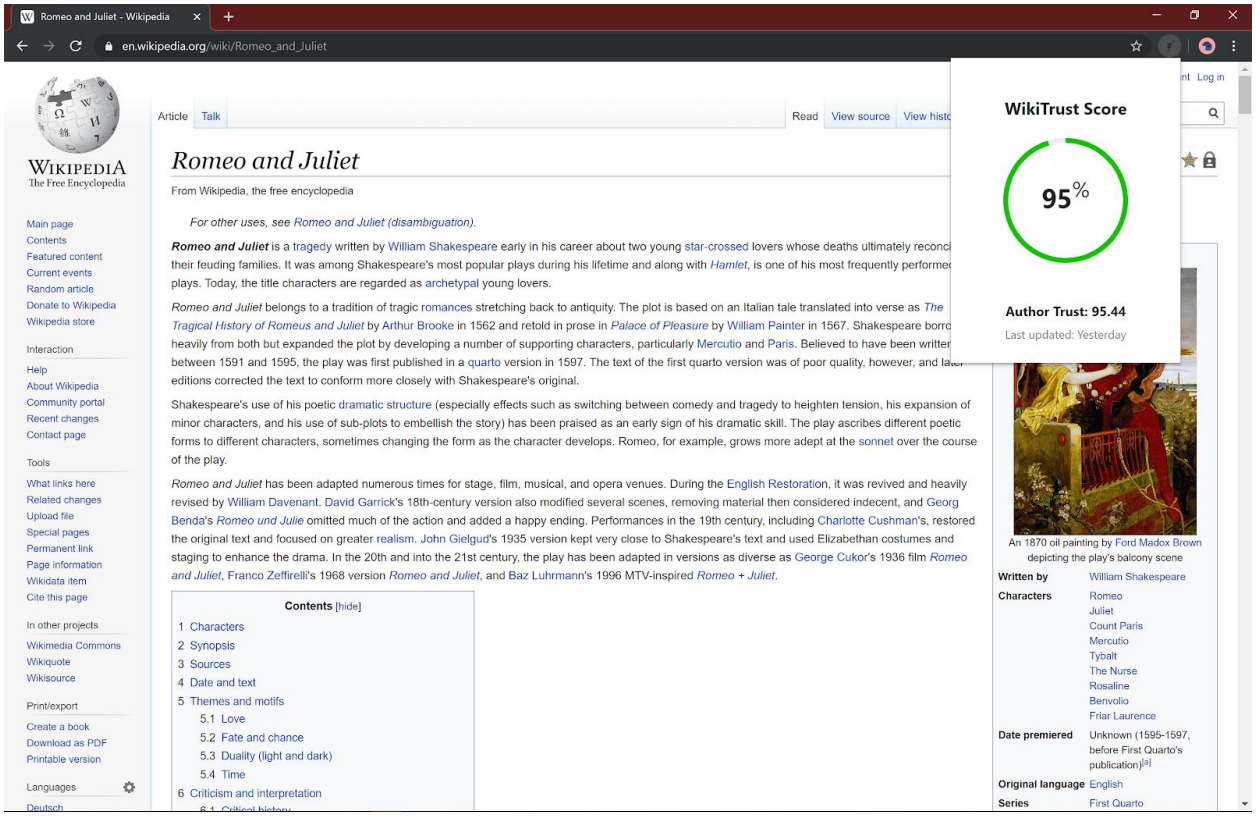
Rev 2.0 - 12/10/19

Goals

- 1-click Chrome/Firefox Extension
- Rank and score pages with an overall “trustworthiness” score from 0 to 100
- Adjust the color based on the computed trust value
- Highlight specific lengths of text and score those
    - A whole page will have different colors
- Also give an “Author Reputation” score to the version.

Image of the MVP





User Stories

Sprint 1:

#1 - As a WikiTrustuser, I would like to see a list of revisions. (5 story points)

Overall time estimate: 20 hours

- Read documentation and familiarize with Wikipedia API. (3 hours)
- Choose the programming language and libraries for the job. (3 hours)
- Write a script to get revisions from Wikipedia API. (10 hours)
- Clean data coming from Wikipedia API script. (4 hours)

#2 - As a WikiTrust user I would like to know that the algorithm assigning a trust value to Wikipedia pages is reliable, so that I know the information I am receiving is reliable. (8 story point)

Overall time estimate: 14 hours

- Study the research papers to understand the algorithms.(10 hours)
- Study the algorithm to understand how it is supposed to work.(3 hours)
- Study the algorithm and decide the input and output.(1 hour)

#3 - As a WikiTrust user, I would like the webpage to load fast + smoothly, so that I have an enjoyable experience (5 story points)

Overall time estimate: 12 hours

- Determine the best cloud host for our project. (3 hours)
- Decide what format our data should be written in and where. (5 hours)
- Research what is the best way to send data between programs. (4 hours)

Sprint 2:

#1 - As a WikiTrust user, I would like the computation to be done in the cloud, so that I do not have to have a powerful computer. (20 story points)

Overall time estimate: 26 hours

- Create an AWS instance and initialize databases. (6 hours)
- Create database python wrapper for easy reads/writes. (7 hours)
- Create a hashing algorithm to store unique diffs. (2 hours)
- Optimize queries to be fast and efficient. (3 hours)
- Write a function that searches a Wikipedia page using a string. (3 hours)
- Write a function that includes extra parameters in queries to get categories, timestamps, and other metadata. (2 hours)
- Write a script that outputs a combination of the previous revision and the current revision of an article. (3 hours)

#2 - As a WikiTrust user, I would like to see a calculated trust value for parts of the Wikipedia page I choose, so that I know how reliable they are individually. (13 story points)

Overall time estimate: 11 hours

- Translate the previous algorithm from python 2.7 to Python 3. (1 hour)
- Test the algorithm to make sure it is working properly. (4 hours)

- Create an AWS instance and run the algorithm on that instance. (5 hours)
- Update the algorithm to receive the input and provide the output. (1 hour)

Sprint 3:

#1 - As a WikiTrust user, I would like to see a Wikipedia article, so that I can use it as a Wikipedia replacement. (8 story points)

Overall time estimate: 7 hours

- Display simple html text in a webpage (2 hours)
- Set up correct routing so pages are the same (2 hours)
- Pull data from wikipedia api for plain article text & display it (2 hours)
- Create an algorithm to display the most relevant page(1 hour)

#2 - As a WikiTrust user I would like to see an overall calculated trust value for the Wikipedia page I choose, so that I know how reliable it is. (13 story points)

Overall time estimate: 18 hours

- Study the previous trust assigning algorithms.(4 hours)
- Write an algorithm to calculate the trust value.(6 hours)
- Write an algorithm to calculate the overall trust value of an article.(3 hours)

#3 - As a WikiTrust user, I would like to see the Wikipedia page I choose colored based on the calculated trust value, so that I can visually understand how reliable that information is. (13 story points)

Overall time estimate: 20 hours

- Create api for grabbing text difference from our api. (10 hours)
- Read + parse what chunks to color. (5 hours)
- Inject color to webapp + color on a yellow->red. (5 hours)

Sprint 4:

#1 - As a WikiTrust user, I would like to use Wikipedia normally, so that I can access the information I want conveniently, and without being distracted. (13 story points)

Overall time estimate: 17 hours

- Write a chrome extension to be able to access the page. (2 hours)

- Have a chrome extension inject code into the Wikipedia page. (10 hours)
- Display calculated trust on the Wikipedia page in a non-intrusive manner. (5 hours)

#2 - As a WikiTrust user, I would like to use a chrome extension, so that I can see reputation scores embedded on the page (13 story points)

Overall time estimate: 28 hours

- Transfer react code into the chrome extension engine. (6 hours)
- Create another DB wrapper for the chrome extension. (3 hours)
- Allow new pages to be crawled upon request (10 hours)
- Write an algorithm that connects the components of the completed algorithms in order to calculate the trust value of a text. (6 hours)
- Re-write the algorithm to calculate the overall trust value of an article to make it compatible with the final algorithm.(3 hours)

Product Backlog

- Graph of affected articles
- Certified edits/pages
- Highlight specific lengths of text and score those

Project Presentation

Presentation can be found here

Final Presentation

Final Presentation can be found here