CS676 Credibility Formulas for Innate Domain Trust and Outgoing Link Trust One Pass Chatbot Webscraper

Joshua Gottlieb

February 17, 2025

1 Introduction and Assumptions

The amount of misinformation present on the internet is enormous. It is necessary to be able to assess the credibility of webpages returned by a Google search in a programmatic way to assist users in understanding how much they can trust each of the returned results. This write-up is meant to detail a potential mathematical foundation for creating a way to assign "star ratings" from 1 to 5 to each returned result based on the links present on that webpage.

There are a number of premises that are assumed for the rest of this document.

1. The results of a web search pertain to a specific domain. It is assumed that there are a number of trusted sites in any particular domain which can serve as the basis for rating all other webpages. For example, in the medical domain, the Centers for Disease Control (CDC) or World Health Organization (WHO) are entities which are assumed to be extremely trustworthy, and therefore, any sites affiliated with these entities is automatically assigned a star rating. Trusted websites should have a rating between 2.5 and 5 stars based on domain knowledge. All other webpages are assumed to have the minimum rating of 1 star. The vast majority of webpages are unknown, and thus, a sufficiently large number of these trusted, known webpages must be provided for this algorithm to produce any meaningful results. These predetermined ratings may be supplied by an API to a service such as Moz and are assumed to be convertible to a scale from 1 to 5. For the purposes of this project, a small selection of websites are curated and semi-arbitrarily assigned star values to use as proof-of-concept.

2. The web scraper assesses only the links present on the webpage. This is because the actual content of a page can take many forms and comprises a complex Natural Language Processing (NLP) task. It is assumed that the links present on a webpage are relevant to the content; however, in reality, there is no guarantee that the links on a webpage correspond to the content on a page whatsoever. This means that any star rating assigned by the algorithm described in this document cannot be used to describe the strength of the content of the page.

3. It is assumed that citations and references are formatted in a way that contains parseable links to known webpages. This may not be the case, as in some forms of academic citation, links are not included or are hidden, such as in the form of Digital Object Identifiers (DOI).

4. The algorithm described in this document is not recursive and only works on a single pass without rating the quality of links on a webpage. Thus, links to unknown webpages are ignored, as the quality of those links cannot be quantified. Only links from known webpages have innate quality. Even in the case of known webpages, webpages that are weak should not be considered useful for boosting credibility, and thus only webpages with a minimum of 2.5 stars should be considered as useful contributors.

5. A webpage could spam many links in order to increase its connectivity in an abritrary and artificial manner. This could inflate the star rating that webpage receives, although the algorithm attempts to reduce the effectiveness of this strategy.

6. An unknown webpage should not exceed the strength of its sources, and thus a final check is needed to truncate the final star rating, if needed.

2 The Star Rating Function

2.1 Function for unknown webpages

The function $f(c)$ for creating a star rating for an unknown webpage is described by:

$$f(c) = 4 \tanh\left(\frac{c}{500}\right) + 1$$

where c is the total contribution score given by the strength of all links included on the webpage. This function returns values in $[1, 5]$ for all nonnegative c. There are several reasons to use the hyperbolic tangent function for generating star ratings.

1. The hyperbolic tangent function grows fairly quickly for "small" inputs but features strong asymptotic behavior, curbing further growth. This naturally mimics the intuitive psychological assumptions surrounding star ratings. Intuitively, it should be easy to acquire the first few stars, but achieving high ratings (> 4) should be progressively more difficult.

2. By using a function with asymptotic properties, it is possible to account for any number of links on a webpage while keeping the output within the appropriate bounds. It is ideal that an unknown webpage would struggle to be more credible than its strongest sources, and it should be nearly impossible for an unknown page to achieve maximum credibility (5) since that rating should be reserved for completely trustworthy expert primary sources.

3. By using a closed form function, calculations are simplified and an inverse function can be solved for algebraically to determine the ideal contribution score needed for a given star rating.

Compare $f(c)$ to a similarly scaled sigmoid $s(c)$ or linear function $\ell(c)$:

$$s(c) = \frac{-3 + 5e^{x/500}}{1 + e^{x/500}}, \qquad \ell(c) = \frac{x}{500} + 1$$

$f(c)$ grows quickly early on, but its rate of growth slows dramatically as c grows, mim-

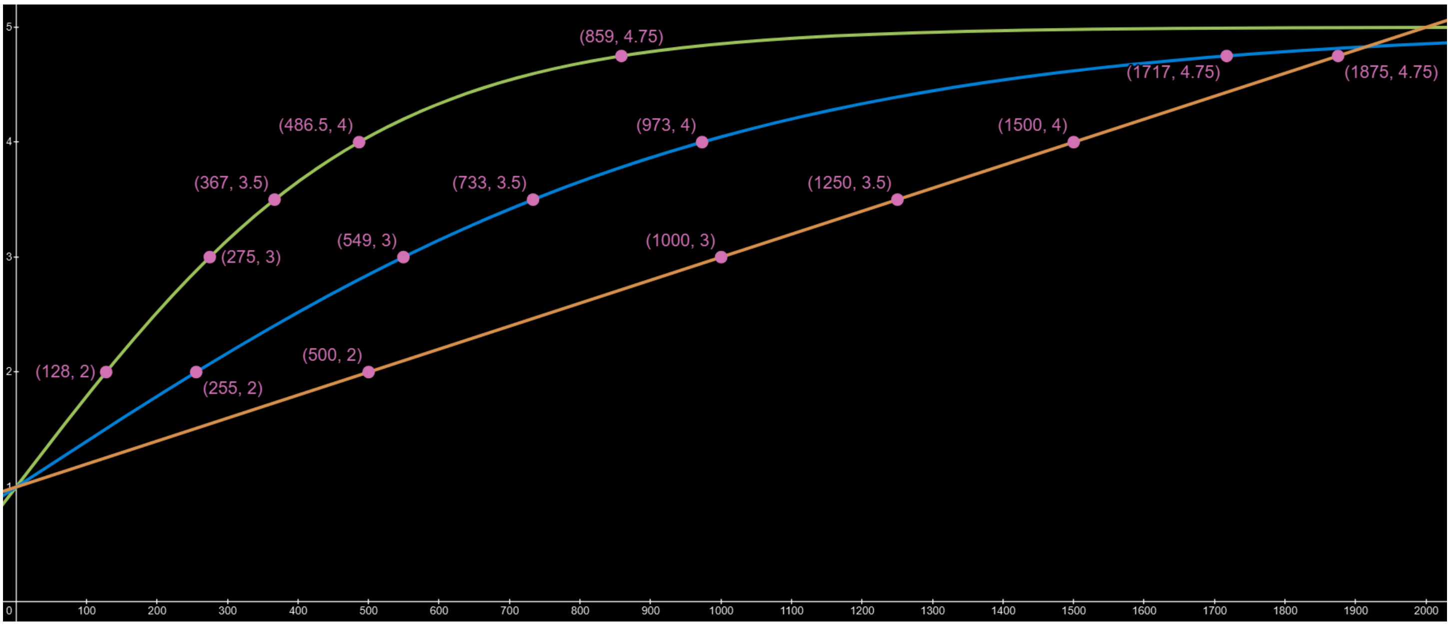

Figure 1: Comparison of a hyperbolic tangent solution $f(c)$ (green), a sigmoid solution $s(c)$ (blue), and a linear solution $\ell(c)$ (orange).

icking the intuitive understanding that early stars are easier to acquire than later stars. $f(c)$ has a convenient closed-form solution for its inverse, making the calculation of contribution breakpoints trivial.

$s(c)$ grows too slowly for low values of c under the given coefficients, making it difficult to acquire any credibility. In addition, the asmpytotic behavior takes too long to kick in. The contribution difference between $s(3.5)$ and $s(4)$ is about the same as the difference between $s(3)$ and $s(3.5)$. Although the coefficients could be changed to modify the shape of the curve, $s(c)$ does not have a closed form solution for its inverse, which complicates the ability to calculate contribution breakpoints.

$\ell(c)$ is too simplistic and does not match to the intuition that later stars should be more difficult to attain than earlier stars. Since the function is linear, it also exhibits the undesirable quality that it is unbounded, which makes it difficult to convert an abritrary number of contribution links into a star rating, something which is avoided by $f(c)$ and $s(c)$ by their bounded asymptotic behavior.

2.2 Inverse star function and adapting the star function for known webpages

Solving for the inverse function $g(s)$ allows for the calculation of a contribution score based on a star rating s.

$$g(s) = f^{-1}(c) = 500 \tanh^{-1}\left(\frac{s-1}{4}\right)$$

Looking at some common star values produces the following table:

stars	contribution score
1	0
2	127.706
3	274.653
4	486.478
4.5	677.013
4.9	1092.362
4.99	1670.840

Using these base values makes it possible to adapt the original star function to incorporate known webpages. By converting the known star rating for a webpage to a contribution score, initial credibility can be accounted for in the rating formula. By assuming unknown webpages start at the minimum 1 star rating ($g(1) = 0$), the full star rating function $h(c, s)$, where s refers to the base star rating of the webpage, becomes:

$$h(c, s) = 4 \tanh\left(\frac{c + g(s)}{500}\right) + 1$$

The final piece of the puzzle is to figure out how much contribution score each link should generate in order to create a reasonable and logical way of transferring the strength of linked sources to the current webpage.

3 Generating Contribution Score from Links

3.1 Contribution sum function for multiple links of same strength

The base contribution function $c(x, b)$ where x is the number of links of the given strength and b is the base contribution for a single link of that strength is described by:

$$c(x, b) = \sum_{n=1}^{x} b^{1-(2(n-1)/21)}$$

This function utilizes a decaying exponent to reduce the additional contribution provided by successive links. This is done for a number of reasons:

1. Intuitively, the largest jump in credibility is achieved from the first source included. A webpage with no sources cited has nothing on which to base its credibility. Additional sources do add to credibility, but realistically, most visitors to a webpage are unlikely to care about more than the strongest 2-3 sources. Therefore, the first few sources are the most influential sources for improving credibility of a webpage.

2. Because these formulas are based on a simplistic webscraper which naively collects all possible links on a webpage without actually analyzing the content of the webpage, it is possible for a webpage to artificially boost its credibility by simply including a large number of links in the body of the page. By decaying the contribution for each link, this process of spamming links provides diminishing returns.

By the 12th link, the exponent is < 0, meaning the overall contribution is negligible, providing a natural cutoff for how many links can contribute to improving the credibility of a webpage.

3. Using this decaying exponent in combination with the hyperbolic tangent star rating function and deliberate choice of base contribution value b makes it difficult for an unknown webpage to surpass the credibility of its strongest source.

3.2 Defining reasonable baseline contribution scores for different star ratings

When choosing a baseline contribution score for a given star rating, there are two favorable and arbitrary conditions imposed which are ideal and are based in intuition:

1. If only one source of a given star rating is present on an unknown webpage, the resultant star rating should be approximately 1.5 stars below its source. This is because a page with only one source is generally not very credible, even if its one source is highly credible. Using a variety of sources ensures the information within is unlikely to be biased or incorrect.

2. If ten sources of a given star rating are present on an unknown webpage, the resultant star rating should be approximately equal to that star rating. With ten sources, it is safe to assume the information is strongly supported and that the webpage is at least as trustworthy as the strength of its sources.

Using these loose rules is the reason for the decaying exponent of $c(x, b)$ being equal to $2(n-1)/21$, as this fraction generates values that follow these ideal rules. The base contribution score b for a star rating s can be solved for by solving the equation

$$c(1, b) = b = g(s - 1.5)$$

because by the first assumption above, we want the first source of strength s to lead to a star rating of $s - 1.5$. This only works for sources of strength ≥ 2.5 stars, as $g(2.5 - 1.5) = g(1) = 0$ and smaller inputs spit out negative values which are not useful. Using some common breakpoints, we get the following base contribution for different source star ratings, as well as the resultant star rating for 1 and 10 sources of that star rating.

stars	base contribution score (b)	$f(c(1, b))$	$f(c(10, b))$
5.0	366.58	3.5	4.74
4.5	274.65	3	4.47
4.0	197.11	2.5	4.03
3.5	127.71	2	3.38

For a graphical comparison of the effects of different sources, see the graph below:

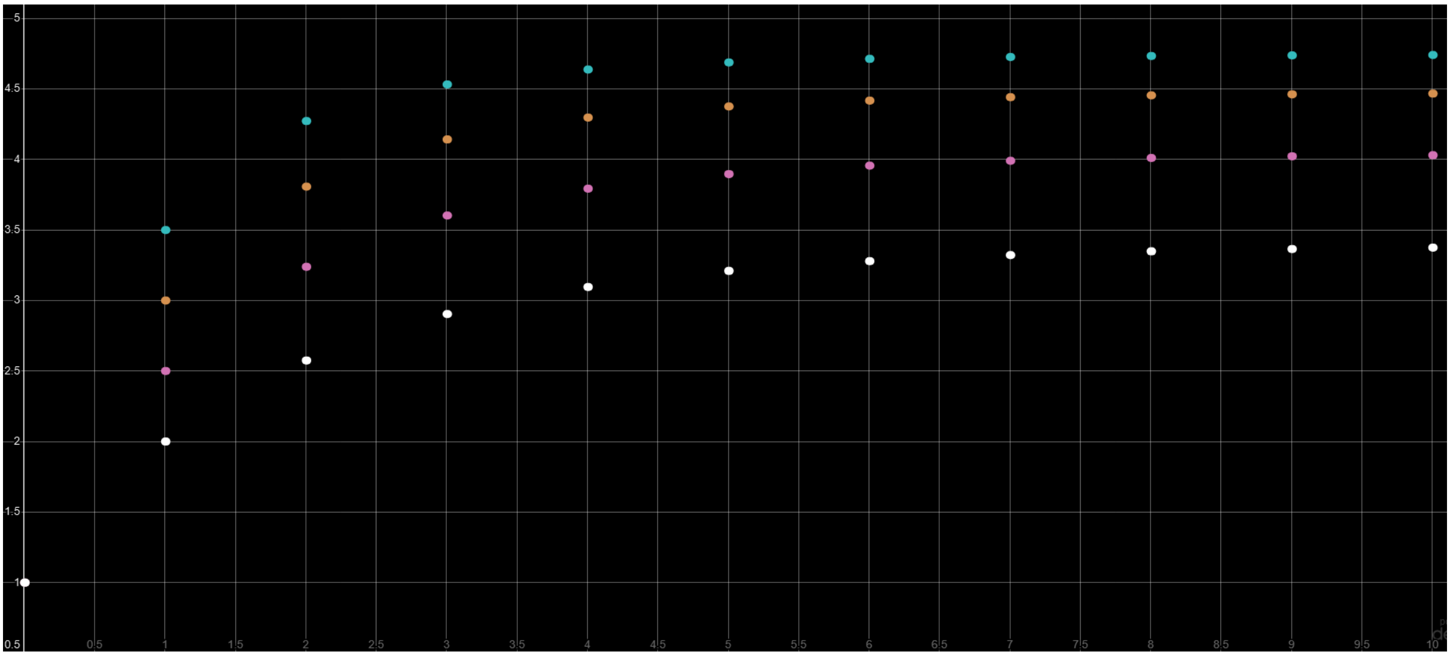

Figure 2: Comparison of end star ratings for x sources of varying star ratings (Blue = 5, Orange = 4.5, Pink = 4, White = 3.5)

As can be seen, the decaying powers and asymptotic nature of $f(s)$ naturally weight the contribution of multiple links to prevent a site from spamming links to artificially inflate its credibility score while also adhering to the intuitive expectation that a reader often cares that a site has at least a few sources before being satisfied that the site is properly credible.

3.3 Extending the contribution sum function for links of differing strengths

The contribution function $c(x, b)$ can be easily extended to accomodate x links of varying strengths by assuming the links present on a webpage are sorted in descending order of strength. Assume that the present links' star ratings $\mathbf{k}$ are arranged as an array in descending order:

$$\mathbf{k} = [k_1, k_2, \dots, k_{x-1}, k_x], \quad k_i \geq k_{i+1}$$

Then, the star ratings can be converted to a contribution array b:

$$\mathbf{b} = [b_1, b_2, \dots, b_{x-1}, b_x], \quad b_i \geq b_{i+1}$$

The contribution function $c(x, b)$ can be rewritten as $c(x, \mathbf{b})$:

$$c(x, \mathbf{b}) = \sum_{n=1}^{x} b_n^{1-(2(n-1)/21)}$$

In other words, the links are arranged in descending order based on strength and the decaying exponent is carried forward for all links, regardless of base. The strongest links keep the largest exponents and thus maintain the strongest contribution.

4 A discussion on the coefficients in this document

In the star rating function, there are 3 coefficients a, b, and s:

$$f(c) = a \tanh\left(\frac{c}{s}\right) + b$$

The coefficients a and b are easy to determine, as these coefficients determine the minimal and maximal asymptotes of the hyperbolic tangent function. Since the goal is to produce outputs between 1 and 5 stars, a and b become:

$$b = \text{minimum value} = 1, \quad a = \text{maximum value} - b = 5 - 1 = 4$$

in order to shift and scale the function to have asymptotes at $(0, 5)$ instead of $(-1, 1)$.

The coefficient s scales the value of the input to help control the shape of the function. The selection of $s = 500$ is completely arbitrary and was chosen because it produced roughly the desired shape when manually inspected.

The link contribution function can be calculated if the star rating function $f(c)$ is established using a particular s. As $f(c)$ is invertible, the inverse rating function is:

$$g(x) = s \tanh^{-1}\left(\frac{x-b}{a}\right) = 500 \tanh^{-1}\left(\frac{x-1}{4}\right)$$

The next step is to apply the following rules to generate the link contribution bases and the decaying exponent:

1. The application of one link of strength x to an unknown webpage results in a star rating for that webpage of $x - 1.5$.
2. The application of ten links of strength x to an unknown webpage approximately results in a star rating for that webpage of x.

Applying rule (1) results in the following formula for generating the base contribution value $b(x)$ for any star rating x.

$$b(x) = g(x - 1.5) = s \tanh^{-1}\left(\frac{x - b - 1.5}{a}\right) = 500 \tanh^{-1}\left(\frac{x - 2.5}{4}\right)$$

Using this value for b allows one to "solve" for the decaying exponent d in the link contribution sum function $c(p, b)$ as detailed in 3.1.

$$c(p, b) = \sum_{n=1}^{p} b^{1-d(n-1)}$$

Using rule (2), where $b(x)$ is the base contribution of a link with star rating x:

$$c(10, b(x)) = b(x) + b(x)^{1-d} + b(x)^{1-2d} + \cdots + b(x)^{1-9d} \approx g(x)$$

An exact solution cannot be calculated directly, but a proper gradient descent algorithm could be utilized to get a "close enough" approximation for d. In this document, $d = 2/21$ was arbitrarily chosen for being "close enough" and easily interpretable as a fraction.

Thus, an algorithm can be created for producing the functions in this document:

1. Choose initial values a and b to set the appropriate minimum and maximum values. Choose a scaling factor s. Together, these create $f(c)$.
2. Calculate the inverse function $g(x)$.
3. Pick a difference δ such that the base contribution $b(x)$ from a link of strength x produces a star rating $x - \delta$ when converted using f. $b(x) = g(x - \delta)$.
4. Decide on an appropriate number of links p_{max} such that $g(c(p_{max}, b(x))) \approx g(x)$. Use gradient descent using a particular base b to calculate the decaying exponent d of $c(p, b)$ by expanding the sum as appropriate, iterating as needed until convergence within an error ε is achieved.

5 Putting it all together into an algorithm

1. Look at the webpage's URL to assess its base star rating. If it is from a known entity, assign the base star rating s_0. If $s_0 = 5$, then the webpage cannot be more credible and the algorithm should return a star rating of 5, citing the source entity's credibility. Otherwise, assign the initial contribution score c_0 as the result of $g(s_0)$.
2. Extract all of the webpage's links and assign star ratings based on API calls in an array $\mathbf{k}$. Drop all star ratings below 2.5, as trusted websites are assumed to have at least a star rating of 2.5. Sort $\mathbf{k}$ in descending order and keep only the first 12 terms, since the exponent decay will effectively eliminate subsequent terms as outlined in section 3.1.
3. Convert the values in array $\mathbf{k}$ to contribution bases in array $\mathbf{b}$. Plug in the values of $\mathbf{b}$ into $c(x, \mathbf{b})$. Because the relevant bases and exponents are known beforehand, these values can be precalculated and loaded into a lookup table for swift processing.
4. Add the contribution score obtained in the previous step to c_0 to obtain the total contribution c. Calculate the final star rating s of the webpage using $f(c)$, then round to the nearest 2 decimals. Since the required contribution scores for star ratings 1.00 to 4.99 can be calculated using $g(s)$, these values can be precalculated and loaded into a lookup table for swift processing without having to actually call any exponential math functions at runtime.
5. Perform a final sanity check: if $s_0 = 1$ (i.e., the webpage is unknown) and s is greater than the largest k, which because $\mathbf{k}$ is sorted, is equivalent to k_1 using 1-based indexing, replace s with k_1. Unknown webpages should never be more credible than their strongest source. Known webpages can exceed their strongest sources because their sources bolster their innate credibility.
6. Return the final calculated value of s with an explanation citing the site's base credibility, the size of $\mathbf{k}$ (number of sources) and maximal star rating in $\mathbf{k}$ (strongest source). Optionally, give more details about the quality of links in $\mathbf{k}$ and the innate quality of the webpage, if applicable.