By Mani Chadaga and Max Christopher

The Page Rank Algorithm:

When a search engine generates a list of pages based on your search result, how does it determine the order it shows you those pages? Obviously, as the viewer, you want to see the most relevant and important websites first. However, this is not as simple as counting the number of times a keyword is mentioned on the site. There must be some other way to judge the importances of each website and then rank them. The Page Rank Algorithm, used by the Google search engine, judges a website's importance by the other sites that link to it. Thus, given a large network of websites, interrelated by their links to one another, the Page Rank Algorithm can assign a relative importance to each page. The sites can then be ranked by these relative importances.

How do Links Translate to Importance?

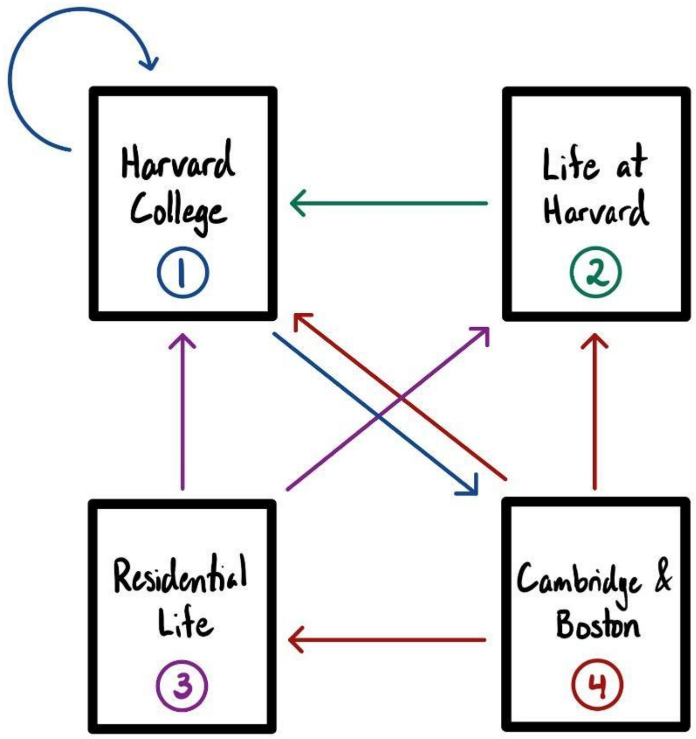
The process for ranking sites in this way is fairly simple. Although it is not inherently obvious how links between sites can lead to rankings and importance, the general theory is that if a site is linked to a lot, it must be important. When many sites link to a certain page, it must mean that the site linked to is largely viewed as credible and relevant. Additionally, when an important site links to another, it shows that this new cite holds enough credibility to be referenced by a popular site.

Thus, given a network, a site can 'transfer' its importance to other sites by linking to them. If a site only links to one other site in the network, it transfers 100% of its importance to that other site. However, if it links to 5 other sites, it transfers ⅕ of its importance to each of them. Simply put, if a site links to k other pages, it will transfer $1/k$ of its importance to each other site.

We can represent this transfer of importance in a transition matrix. In the transition matrix for a network, each column represents a site's outgoing links, and each row represents a site's incoming links. Once we calculate the transition matrix, T, for the given network, we want to calculate the stable state of the system. This means we want to calculate the vector x such that $Tx = x$. In other words, we want to calculate the eigenvector, x, for the eigenvalue 1. This eigenvector will give us the relative importance of each website. To find the total percentage of importance each site has, simply scale x so that the sum of the components equals 1. Then, each component of the eigenvector will represent the percent importance of each site.

Page Rank in Practice:

Let's see how a network of websites are ranked in an example. Take the following network of 4 websites, where an arrow represents links from one site to another:



As shown in the network, site 1 transfers ½ of its importance to itself and site 4. Site 2 transfers all its importance to site 1. Site 3 splits its importance between sites 1 and 2. Finally, site 4 splits its importance into thirds between all the other websites. Thus, we get the following transition matrix:

$$\begin{array}{cc} & \begin{array}{c}\text{Outgoing Links}\\ \text{from Site 1}\\ \downarrow\end{array} \\ \text{Incoming Links}\ \begin{array}{c}\text{Site 1}\\ \text{Site 2}\\ \text{Site 3}\\ \text{Site 4}\end{array} & \begin{bmatrix} \frac{1}{2} & & & \\ 0 & & & \\ 0 & & & \\ \frac{1}{2} & & & \end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{c}\text{Outgoing Links}\\ \text{from Site 2}\\ \downarrow\end{array} \\ \text{Incoming Links}\ \begin{array}{c}\text{Site 1}\\ \text{Site 2}\\ \text{Site 3}\\ \text{Site 4}\end{array} & \begin{bmatrix} \frac{1}{2} & 1 & & \\ 0 & 0 & & \\ 0 & 0 & & \\ \frac{1}{2} & 0 & & \end{bmatrix}\end{array}$$

$$\begin{array}{cc} & \begin{array}{c}\text{Outgoing Links}\\ \text{from Site 3}\\ \downarrow\end{array} \\ \text{Incoming Links}\ \begin{array}{c}\text{Site 1}\\ \text{Site 2}\\ \text{Site 3}\\ \text{Site 4}\end{array} & \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{2} & \\ 0 & 0 & \frac{1}{2} & \\ 0 & 0 & 0 & \\ \frac{1}{2} & 0 & 0 & \end{bmatrix}\end{array} \quad \begin{array}{cc} & \begin{array}{c}\text{Outgoing Links}\\ \text{from Site 4}\\ \downarrow\end{array} \\ \text{Incoming Links}\ \begin{array}{c}\text{Site 1}\\ \text{Site 2}\\ \text{Site 3}\\ \text{Site 4}\end{array} & \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & 0 \end{bmatrix}\end{array}$$

Now, we simply must calculate the eigenvector for the eigenvalue 1. To do so, we start by subtracting the identity matrix from our transition matrix:

$$\begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & -1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & -1 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & -1 \end{bmatrix}$$

Now, we just have to solve for the vector *v* such that (T-I)*v* = 0. We can see this process explicitly through row reduction:

$$\begin{bmatrix} -\frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & -1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & -1 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & -1 \end{bmatrix} \xRightarrow{+\text{Row }1} \begin{bmatrix} -\frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & -1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & -1 & \frac{1}{3} \\ 0 & 1 & \frac{1}{2} & -\frac{2}{3} \end{bmatrix} \xRightarrow{\cdot(-2)} \begin{bmatrix} 1 & -2 & -1 & -\frac{2}{3} \\ 0 & -1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & -1 & \frac{1}{3} \\ 0 & 1 & \frac{1}{2} & -\frac{2}{3} \end{bmatrix} \xRightarrow{\cdot(-1)} \begin{bmatrix} 1 & -2 & -1 & -\frac{2}{3} \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{3} \\ 0 & 0 & -1 & \frac{1}{3} \\ 0 & 1 & \frac{1}{2} & -\frac{2}{3} \end{bmatrix}$$

$$\xRightarrow[-\text{Row }2]{+2\text{Row }2} \begin{bmatrix} 1 & 0 & -2 & -\frac{4}{3} \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{3} \\ 0 & 0 & -1 & \frac{1}{3} \\ 0 & 0 & 1 & -\frac{1}{3} \end{bmatrix} \xRightarrow{\cdot(-1)} \begin{bmatrix} 1 & 0 & -2 & -\frac{4}{3} \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{3} \\ 0 & 0 & 1 & -\frac{1}{3} \\ 0 & 0 & 1 & -\frac{1}{3} \end{bmatrix} \xRightarrow[-\text{Row }3]{+2\text{Row }3,\ +\frac{1}{2}\text{Row }3} \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -\frac{1}{2} \\ 0 & 0 & 1 & -\frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

And then we can easily see that a solution to this equation is:

$$\begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -\frac{1}{2} \\ 0 & 0 & 1 & -\frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ \frac{1}{2} \\ \frac{1}{3} \\ 1 \end{bmatrix} = 0$$

However, because we want a percentage of importance, we want the components of the vector *v* to sum to 1. Thus, we divide the vector v by the sum of its components, leaving us with the vector:

$$\frac{6}{23}\begin{bmatrix} 2 \\ \frac{1}{2} \\ \frac{1}{3} \\ 1 \end{bmatrix} \approx \begin{bmatrix} 0.52 \\ 0.13 \\ 0.09 \\ 0.26 \end{bmatrix}$$

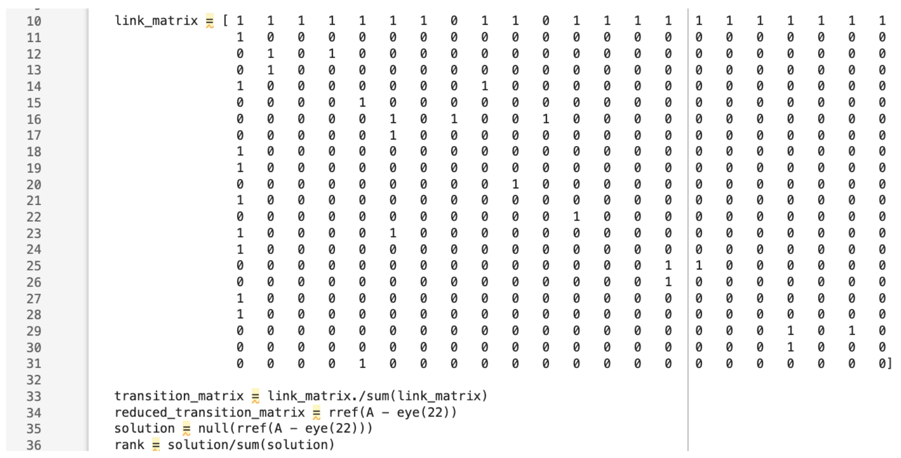
Therefore, we see that for our network, Site 1 has 52% importance, Site 2 has 13% importance, Site 3 has 9% importance and Site 4 has 26% importance. Thus, the Page Rank from most to least important is Site 1, 4, 2, 3.

General Formula for Calculating Page Rank:

1. Compute the Transition Matrix of the Network - denote this matrix T. It is created such that column *i* has 1/*k* in row *j* if site *i* links to site *j* (and site *i* has *k* total outgoing links)
2. Compute the *v* such that (T-I)*v* = 0
3. Divide *v* by the Sum of the Components of *v*
4. The *i*th component of *v* represents the relative importance of Site *i*
5. Rank Sites in Order of Relative Importance

Using MATLAB to Calculate the 22x22 Page Rank:

For our project's example of 22 Harvard College websites, it was not feasible to calculate the Page Rank by hand. Therefore, we used MATLAB to perform calculations for the ranking. First, we created a linking matrix that simple denoted a link between websites with a 1, and no link with a 0:



```
link_matrix = [ 1   1   1   1   1   1   1   0   1   1   0   1   1   1   1   1   1   1   1   1   1   1
                1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                1   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   1   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
                1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
                1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
                1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   1   0
                0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0
                0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]

transition_matrix = link_matrix./sum(link_matrix)
reduced_transition_matrix = rref(A - eye(22))
solution = null(rref(A - eye(22)))
rank = solution/sum(solution)
```

The code below link_matrix completes the following steps:

(1) **transition_matrix=link_matrix./sum(link_matrix)** creates the transition matrix by dividing each value in column *i* by the sum of column *i*

(2) **reduced_transition_matrix=rref(A-eye(22))** calculates the reduced row echelon form of the transition matrix minus the identity matrix

(3) **solution=null(rref(A-eye(22)))** calculates the eigenvector for the eigenvalue 1

(4) **rank=solution/sum(solution)** normalizes the eigenvector such that its components sum to 1

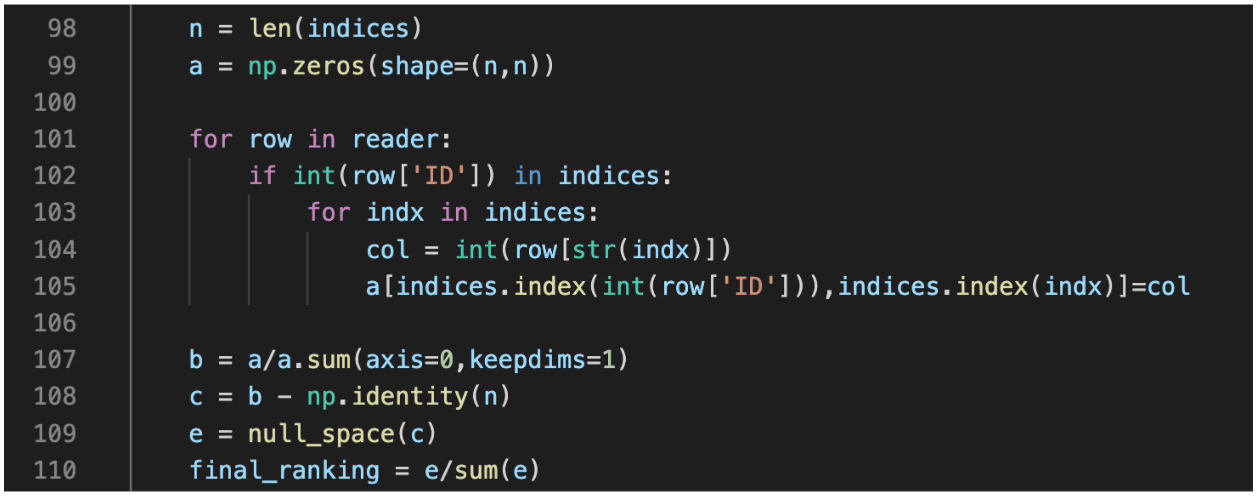
Therefore, the ranking of the 22 Harvard College Site network resulted in the following ranking:

Title	Ranking
Harvard College	0.3974
College Offices	0.0596
Events Calendar	0.0447
Current Students | Harvard	0.0397
Advising	0.0397
Guide to Student Employment	0.0397
A Guide to Finding Funding	0.0397
Study Abroad	0.0397
Student Stories	0.0397
Cambridge & Boston	0.0397
Harvard University	0.0298
Academics	0.0265
Financial Aid	0.0199
Office of Undergraduate Education	0.0199
Student Employment Office	0.0199
Research	0.0199
Life at Harvard	0.0199
About | Harvard College	0.0199
Apply For Financial Aid	0.0132
Liberal Arts and Sciences	0.0132
Residential Life	0.0132
HarvardCollege Student Handbook	0.005

Turning It Into A Search Engine

At this point in our project we had a static website that displayed the ranking of all 22 Harvard College webpages. But Google and other search engines don't display *all* available pages – they display only the pages that fit a certain keyword, and then rank those pages. So we set out to apply our PageRank to a Harvard search engine, and we did so by taking the following steps:

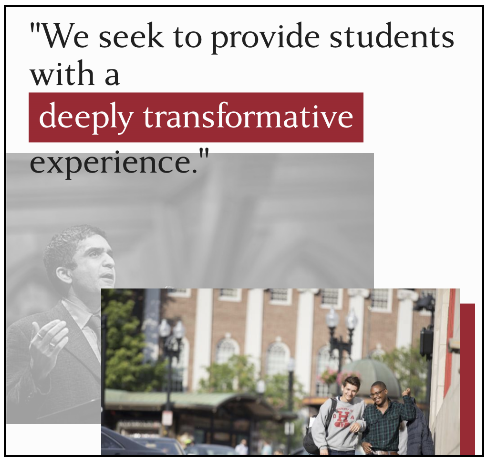
1) Rewriting our PageRank in Python code as opposed to MATLAB

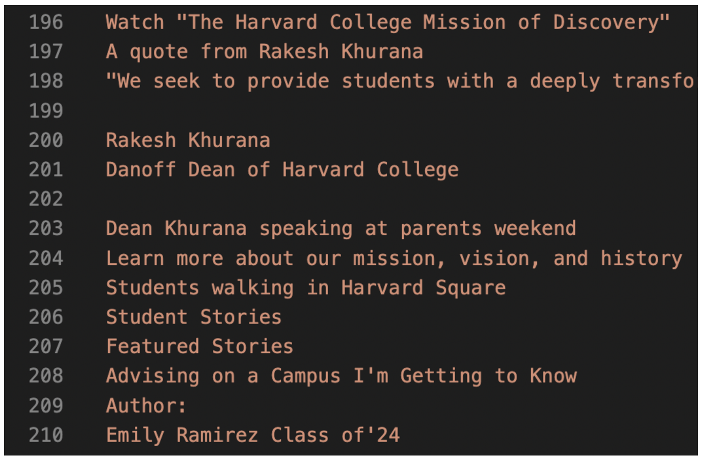


```python
    n = len(indices)
    a = np.zeros(shape=(n,n))

    for row in reader:
        if int(row['ID']) in indices:
            for indx in indices:
                col = int(row[str(indx)])
                a[indices.index(int(row['ID'])),indices.index(indx)]=col

    b = a/a.sum(axis=0,keepdims=1)
    c = b - np.identity(n)
    e = null_space(c)
    final_ranking = e/sum(e)
```

This was fairly simple thanks to NumPy and other linear algebra packages.

2) Scraping the text from each of our 22 Harvard College sites

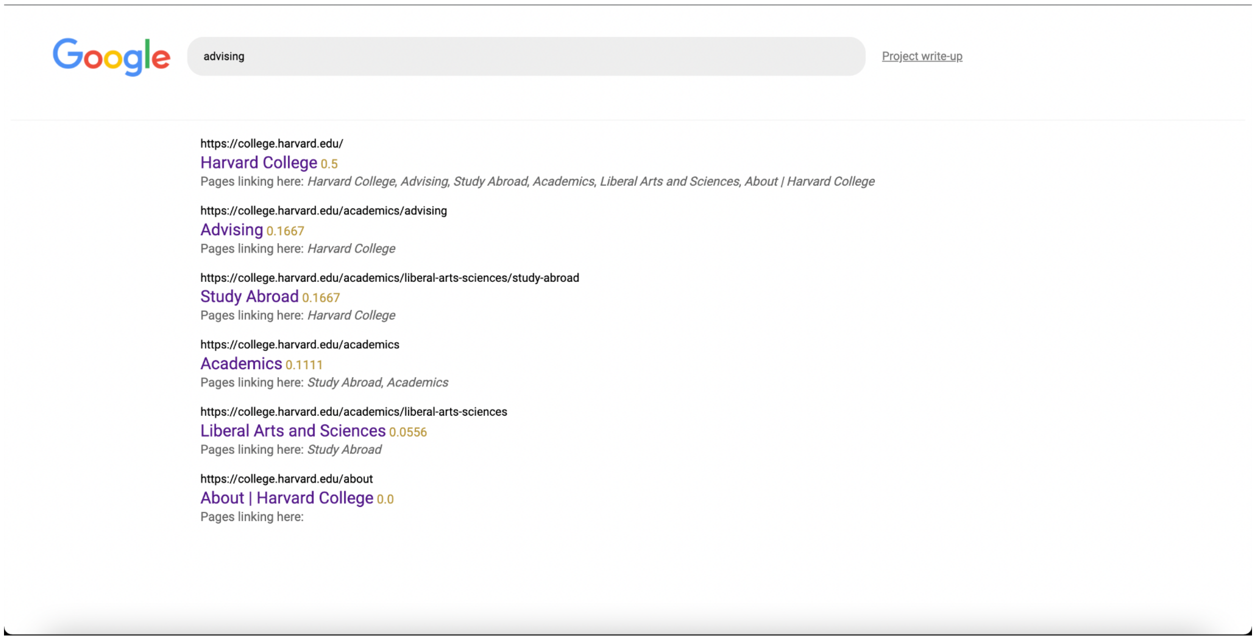




```
Watch "The Harvard College Mission of Discovery"
A quote from Rakesh Khurana
"We seek to provide students with a deeply transfo

Rakesh Khurana
Danoff Dean of Harvard College

Dean Khurana speaking at parents weekend
Learn more about our mission, vision, and history
Students walking in Harvard Square
Student Stories
Featured Stories
Advising on a Campus I'm Getting to Know
Author:
Emily Ramirez Class of'24
```

Because there were few pages, we didn't need any sophisticated infrastructure (like what Google has) to index all the pages in milliseconds.

3) Then, we created a search bar where, once the user types in a keyword, we would filter out which pages' text contained that keyword, then use our PageRank script on only those pages. This worked well for some keywords, but we encountered problems with others. Namely, when there were too few pages matching a keyword, PageRank would fail because there weren't enough links in between the pages. Also, the 'Harvard College' page reigned supreme in nearly all cases, so our next step could be researching how Google prevents popular sites (like Google itself) from taking #1 every time.



https://college.harvard.edu/
Harvard College 0.5
Pages linking here: *Harvard College, Advising, Study Abroad, Academics, Liberal Arts and Sciences, About | Harvard College*

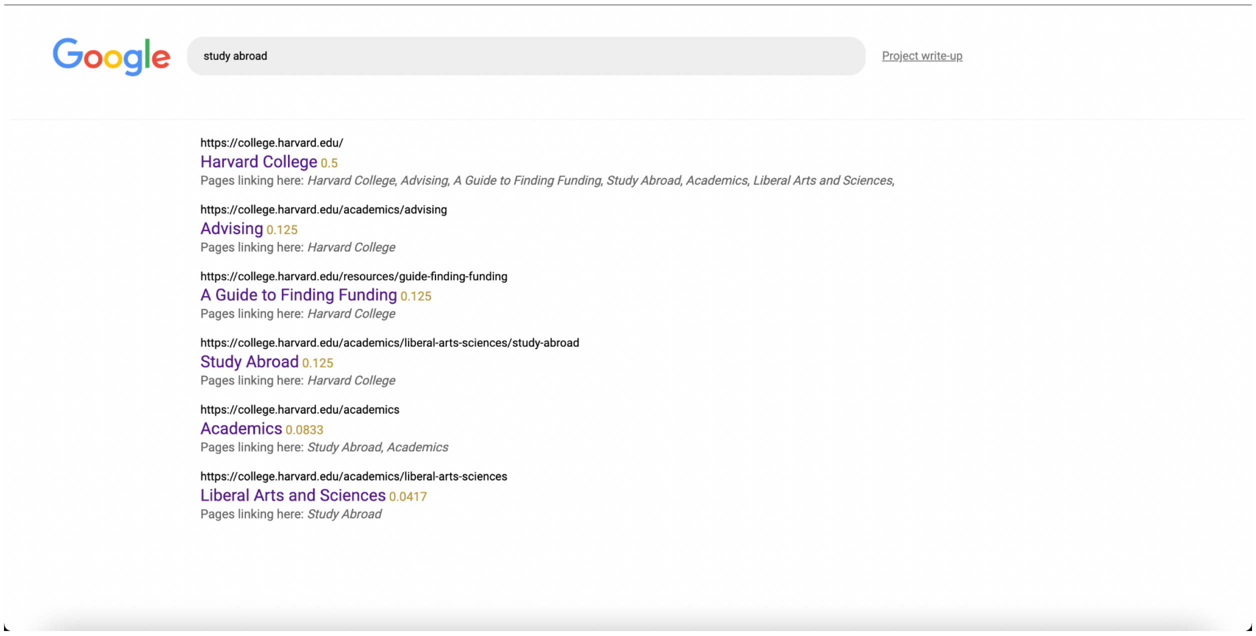
https://college.harvard.edu/academics/advising
Advising 0.1667
Pages linking here: *Harvard College*

https://college.harvard.edu/academics/liberal-arts-sciences/study-abroad
Study Abroad 0.1667
Pages linking here: *Harvard College*

https://college.harvard.edu/academics
Academics 0.1111
Pages linking here: *Study Abroad, Academics*

https://college.harvard.edu/academics/liberal-arts-sciences
Liberal Arts and Sciences 0.0556
Pages linking here: *Study Abroad*

https://college.harvard.edu/about
About | Harvard College 0.0
Pages linking here:



https://college.harvard.edu/
Harvard College 0.5
Pages linking here: *Harvard College, Advising, A Guide to Finding Funding, Study Abroad, Academics, Liberal Arts and Sciences,*

https://college.harvard.edu/academics/advising
Advising 0.125
Pages linking here: *Harvard College*

https://college.harvard.edu/resources/guide-finding-funding
A Guide to Finding Funding 0.125
Pages linking here: *Harvard College*

https://college.harvard.edu/academics/liberal-arts-sciences/study-abroad
Study Abroad 0.125
Pages linking here: *Harvard College*

https://college.harvard.edu/academics
Academics 0.0833
Pages linking here: *Study Abroad, Academics*

https://college.harvard.edu/academics/liberal-arts-sciences
Liberal Arts and Sciences 0.0417
Pages linking here: *Study Abroad*