Chapter 1

Introduction

1.1 PageRank

PageRank is an algorithm which is used by Google for ranking web pages according to their importance. It simply attempts to measure the importance of a web page.

“Link popularity” is the main idea behind this algorithm. A certain page will get a good rank value if it receives more incoming links on it. There are many important factors that determine the ranking of a web page but PageRank remains to be the basic tool for ranking purpose.

1.2 Operation of a Search Engine

The process of PageRank starts with a search. If we know the URL of a specific web page, we can simply type the URL into a search engine and reach our desired website. But what will happen if we do not? Then the search engine comes into play. The World Wide Web is a vast area. A search engine has to find what exactly a person is looking for. We submit a query to the search engine and it does some specific work.

To find out what exactly a person is looking for, a search engine will scan the index of web pages for the contents that match the query. First thing is “Crawling”. A web crawler is one kind of program that is also known as “spider” or “bot” that crawls the web continuously. It looks the find out new web pages on the web and collects information. Information can be anything like images, page title or any keywords.

Next thing a search engine does is that it indexes the result. As the web crawler crawls the web, it looks to find new web pages. The crawler makes a copy of the webpage and adds the web pages URL to an index. After doing so, the crawler follows all the links on the webpage and repeats the process until it reaches the last one. By doing so, it creates a huge index of many web pages. Some web pages may not allow the crawler to crawl them. Then the pages will be left out of the index. Information that a crawler gives to a search engine becomes the search engine’s index. A web page that is suggested by search engine has been crawled by the crawler.

After that, the search engine processes the query. By processing, it means that the search engine sorts out the most relevant answers to the query. PageRank helps in this cause. It helps to find the best result on the basis of most links that point to a page. So, the most important page will appear higher up in the results. The web pages on the first page of a search engine are the ones with the best PageRank value. Many search engine may use different techniques or algorithms. They may give a different result based on their technique. Google uses PageRank as its base. A trustworthy domain also plays an important role in processing a query.

1.3 Basic Intuition of PageRank

The basic intuition about exploring the web is simple. A random surfer starts at any web page and he visits the links on that particular web page. If at any time the surfer gets bored or gets to such link where there is no way to get to a new webpage, the surfer makes a random jump from that web page and ends up being on a whole new web page. This is how the idea of PageRank algorithm comes. PageRank algorithm gives a Probabilistic output that how likely a surfer randomly clicks on links and will arrive at a particular web page. As PageRank algorithm analyses the link structure of the web it assigns a numerical value to each web pages. A PageRank gets its numeric value from a mathematical algorithm based on the web graph. The value obtained after ranking indicates the importance of a particular page. Links to pages are counted as a poll of support. The PageRank algorithm is recursive. It depends on the number of incoming links to a webpage. A web page which is linked by many other web pages gets a high rank itself.

1.4 Factors

A number of links pointing to a web page is an important factor for PageRank. But by creating a web page and getting randomly linked by web pages does not really help to get a good PageRank score. Also referring to other web pages randomly does not help also. Then how to achieve good PageRank values? There are certainly some good ways to achieve a good PageRank score.

PageRank gives a page a rank value on a scale of 1 to 10. So, higher value means that the page is important. All incoming links to a web page do not have the same importance. A web page with only one incoming link can have a higher PageRank value than a web page which might contain a lot of incoming links. This is because if the only incoming link holds a much higher PageRank value than so many other incoming links. As the incoming links work like a poll, they pass their importance to other web pages. So, having a lot of links does not mean that a web pages rank value will increase overnight but it is all about the quality of the incoming link. If quality links are managed and got referred then the website will definitely be ranked higher in a search engine.

There are some other vital factors rather than quality incoming links. To take a web page high up in a search engine some necessary action is needed to be taken into account. The number of visitors to a website is needed to be increased. As this is related to a search engine and optimization is necessary. This technique is called “Search engine optimization” (SEO). A crucial feature of SEO is to make a website easily discoverable for users and search engine spiders.

The search engine optimization (SEO) of a website depends on how quickly and efficiently someone finds that particular website through a search engine query. If the PageRank value is higher the web page of the website will be found easily as the search result will put the web pager higher on the list. There are more than 200 factors that help build a very optimized website. Websites with an effective SEO strategy can get a good rank value. Some strategies are introduced below.

There are two types of SEO. First one is “On-page SEO” and the other one is “Off-page SEO”.

- In the case of On-page SEO, the practice is to optimize individual webpages to get a good PageRank score. Some strategies like starting the title tag with some target keywords to attract the users. Specific keyword title helps to get more weight on Google. Putting keyword inside first 100 words is a good practice. Meta tags for websites are crucial. Good descriptions can attract web surfers. Using some quality outlinks can make the webpage more weighted and by manipulating some links good rank scores can be achieved. So, there are more On-page techniques like them. Using these techniques can really help on optimization.

- In case of Off-page SEO, the strategies are very clear. Like staying updated with Googles algorithm for ranking. Updates are available in time and the updates can be used. Getting high-quality links is a top strategy. It is also necessary to fix broken links if there is any. Because sometimes some links do not stay active. So removing the broken or unused links are important.

1.5 Motivation and Purpose

After the World Wide Web invented by late 1990’s, the amount of content on the web has simply exploded. Searching for necessary content has become a challenging factor. As of the latest information, the web contains a mammoth 4.7 billion of web pages and all of them are indexed. The number is ever growing. So, a method is absolutely necessary today for extracting the proper information from our query.

Our main purpose to study PageRank algorithm is to increase the rank of a web page effectively so that we can display the web page on the front of any search engine when a search query is received related to the webpage.

In this thesis paper, as a sample, a mini web structure is considered. The nodes in the graphs are considered as webpages and the connection among the nodes are the links that connect one another. By using the PageRank formula the corresponding equations for all the nodes are determined. Then the rank values are calculated. The mini web is represented as a matrix structure. For the calculation of rank values “Power Method” is used. To use power method, “Stochastic matrix” is used. A stochastic matrix is a square matrix which describes the transition of “Markov chain”. Markov chain is simply a process that satisfies Markov property and it can make future predictions based on current state. Entries in the matrix are nonnegative real numbers, each of which indicates a probability. Some entry in the matrix may not contain any value, though the matrix we are working with, each entry contains a value. Again, the stochastic matrix can be either row or column stochastic. The stochastic matrix is a very helpful idea to find out the rank values of all webpages in the graph. But the stochastic matrix will face failure if the webgraph contains such node or nodes which do not contain any outgoing link. Such nodes are called “Dangling node”. Because of this dangling node problem, some of the web pages ranks might not be calculated properly or at all.

In order to solve the dangling node problem, a new kind of matrix is proposed. It is known as the “Google matrix”. It uses a new formula that creates such a matrix that contains all nonnegative entries. The advantage of using such matrix is that rank values for each webpage can be measured. But for this advantage, there is a cost. The real word web is huge. For this gigantic graph structure, the amount of physical memory needed is also gigantic. So, the Google matrix also becomes inefficient.

As the measurement of PageRank value is iterative, at least 50 iterationsare considered to get the proper result. It is a common practice.

Chapter 2

Background Study

2.1 Algorithms

A step-by-step procedure or set of rules to be followed for solving a problem or accomplishing some end, especially by a computer.

2.1.1 PageRank Algorithm

The famous PageRank algorithm was invented by Larry Page and Sergey Brin. It was first developed at Stanford University in the year of 1996. The algorithm was named after Larry Page. At the very beginning, it gave the idea of a new kind of search engine. The research project started in 1995 and finally led to a functional prototype in 1998. PageRank has exclusive right assigned to Stanford University only. Google has exclusive license rights on the exclusive right that is assigned to Stanford University.

- **PageRank Equation**

PageRank is defined by Larry Page and Sergey Brin as “We assume page A has T1…Tn which point to it (i.e., are citations). The parameter “d” is damping factor which can be set between 0 and 1. We usually set d to “0.85”. Also, C (A) is defined as the number of links going out of page A”. The PageRank of A is given as follows:

$$PR(A) = (1-d) + d\,(\frac{PR(T1)}{C(T1)} + \frac{PR(T2)}{C(T2)} + \ldots + \frac{PR(Tn)}{C(Tn)})$$

As PageRank is a probability distribution over web pages, so the sum of all web pages PageRank will be one.

- **Damping Factor**

The damping factor d, which is the click-through probability, is included to prevent sinks (i.e. pages with no outgoing links) from "absorbing" the PageRanks of those pages connected to the sinks. It is easy to see that an infinite surfer would have to end up in a sink given enough time, so the damping factor allows a heuristic to offset the importance of those sinks. That is why the first term of the PageRank equation, (1−d), is included. It is the chance of being on a random page after the start, while the second term is normalized so that all PageRanks sum to one.

If d=1, the person clicking will click forever and will always end up in a sink. In this case, the first term is discarded. The second term, given an infinite number of iterations to convergence, is equivalent to finding the steady state of the Markov chain representing pages and links.

On the other hand, if the click-throughprobability is d=0, then all clicks are random restarts. So, a damping factor 0<d<1 is a sort of weighted average between two extremes.

- **Updated PageRank Equation**

The formula stated above has a problem. It does not converge to 1. So, there is another formula which is supported correctly. It is stated below:

$$PR(A) = \frac{1-d}{N} + d\ (\frac{PR(T1)}{C(T1)} + \frac{PR(T2)}{C(T2)} + \ldots + \frac{PR(Tn)}{C(Tn)}$$

Where “N” denotes the number of pages.

Page and Brin confused the two formulas in their most popular paper "The Anatomy of a Large-Scale Hyper textual Web Search Engine", where they mistakenly claimed that the former formula formed a probability distribution over web pages. Google recalculates PageRank scores each time it crawls the Web and rebuilds its index. As Google increases the number of documents in its collection, the initial approximation of PageRank decreases for all documents. If a page has no links to other pages, it becomes a sink and therefore terminates the random surfing process. If the random surfer arrives at a sink page, it picks another URL at random and continues surfing again [1].

2.1.2 Adaptive PageRank Algorithm

In this paper, it was observed that the convergence pattern in the PageRank algorithm has a ununiform distribution. Especially many webpages converge to their true rank values quickly while few pages take much time to converge. So, a new kind of PageRank algorithm was proposed named as “Adaptive PageRank” which speeds up the computation by 30%.

2.1.3 Necessity of Adaptive PageRank

PageRank computes the principal eigen vector of the matrix describing the links on the web using the power method. Due to the sheer size of the web, this computation can take several days. Speeding up the computation is important for two reasons. Firstly, computing PageRank quickly is necessary to reduce the lag time from when a new crawl is completed to when the crawl can be made available for searching. Secondly, personalized and topic-sensitive PageRank schemes are biased towards a certain type of pages. These approaches intensify the need for faster approaches for computing PageRank.

Topic-Sensitive PageRank (commonly referred to as TSPR) is a context-sensitive ranking algorithm for web search developed by Taher Haveliwala while at Stanford University and thought to be used by Google for the purpose of indexing and ranking search results in the search engine results pages, although no evidence has been shown of it in practice [2].

2.1.4 Reason behind Adaptive PageRank

The intuition behind the adaptive algorithm was the reduction of redundant iterations. In other words, there is no need to compute the PageRank values of the pages that already converged. Here Power method and the matrix are used for rank computation. Power method is an iterative approach. So,after some iteration, if some webpages that are found to have converged, the matrix is split into two submatrices where one matrix contains the converged pages and the other matrix contains the pages that have been converged not yet. By doing so, the converged matrix will no longer be necessary for further more computation. So, redundancy can be removed here. It is also observed that slow converging pages are generally pages with higher Pagerank value.

2.1.5 HITS Algorithm

Hyperlink-Induced Topic Search (**HITS**; also known as **hubs and authorities**) is a link analysis algorithm that rates Web pages, developed by Jon Kleinberg. The idea behind Hubs and Authorities stemmed from a particular insight into the creation of web pages when the Internet was originally forming; that is, certain web pages, known as hubs, served as large directories that were not actually authoritative in the information that they held, but were used as compilations of a broad catalog of information that led users direct to other authoritative pages. In other words, a good hub represented a page that pointed to many other pages, and a good authority represented a page that was linked by many different hubs [3].

The scheme, therefore, assigns two scores for each page: its authority, which estimates the value of the content of the page, and its hub value, which estimates the value of its links to other pages.

In the HITS algorithm, the first step is to retrieve the most relevant pages to the search query. This set is called the *root set* and can be obtained by taking the top pages returned by a text-based search algorithm. A *base set* is generated by augmenting the root set with all the web pages that are linked from it and some of the pages that link to it. The web pages in the base set and all hyperlinks among those pages form a focused subgraph. The HITS computation is performed only on this focused subgraph. According to Kleinberg the reason for constructing a base set is to ensure that most (or many) of the strongest authorities are included.

Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values of the pages it points to. Some implementations also consider the relevance of the linked pages.

The algorithm performs a series of iterations, each consisting of two basic steps:

- **Authority Update**: Update each node's *Authority score* to be equal to the sum of the *Hub Scores* of each node that points to it. That is, a node is given a high authority score by being linked from pages that are recognized as Hubs for information.
- **Hub Update**: Update each node's *Hub Score* to be equal to the sum of the *Authority Scores* of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node are calculated with the following algorithm:

- Start with each node having a hub score and authority score of 1.
- Run the Authority Update Rule
- Run the Hub Update Rule
- Normalize the values by dividing each Hub score by square root of the sum of the squares of all Hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores.
- Repeat from the second step as necessary.

HITS, like Page and Brin's PageRank, is an iterative algorithm based on the linkage of the documents on the web. However, it does have some major differences:

- It is query dependent, that is, the (Hubs and Authority) scores resulting from the link analysis are influenced by the search terms;

- As a corollary, it is executed at query time, not at indexing time, with the associated hit on performance that accompanies query-time processing.
- It is not commonly used by search engines
- It computes two scores per document, hub and authority, as opposed to a single score;
- It is processed on a small subset of ‘relevant’ documents (a 'focused sub graph' or base set), not all documents as was the case with PageRank.

HITS *(*hyperlink-induced topic search*)* is now part of the **Ask** search engine (www.Ask.com) [1].

2.1.6 Weighted PageRank Algorithm

The Weighted PageRank algorithm (WPR), an ex-tension to the standard PageRank algorithm. WPR takes into account the importance of both the inlinks and the outlinks of the pages and distributes rank scores based on the popularity of the pages.

The more popular web pages are the more linkages that other webpages tend to have to them or are linked to by them. The proposed extended PageRank algorithm–a Weighted PageRank Algorithm–assigns larger rank values to more important (popular) pages instead of dividing the rank value of a page evenly among its outlink pages. Each outlink page gets a value proportional to its popularity (its number of inlinks and outlinks). [4]

2.2 URL

A Uniform Resource Locator (URL), colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. URLs occur most commonly to reference web pages [5].

2.3 Search Engine before Google (WWWW)

The link structure of the web was one of the main focus. Google is designed in such way to crawl and index the web efficiently and produce a satisfying result on every query it receives. Web search engines before Google, for example, in 1994 one of the first search engines were World Wide Web Worm (WWWW) had indexed 110,000 webpages.The World-Wide Web Worm (WWWW) is claimed to be the first search engine for the World-Wide Web, though it was not released until March 1994, by which time a number of other search engines had been made publicly available. It was developed in September 1993 by Oliver McBryan at the University of Colorado.

The worm created a database of 300,000 multimedia objects which could be obtained or searched for keywords via the WWW. In contrast to present-day search engines, the WWWW featured support for Perl, regular expressions.

In 1994, it received on average 1500 queries per day. Later on 1997, it claimed to have handled 20 million web queries per day. It can be observed that the web was expanding madly. So, efficiency was needed both in quality and scalability.

To meet all the needs to improve efficiency a search engine was needed that could crawl faster and gather more information quickly. Storage space must be needed to use effectively as well. The indexing system must process hundreds of gigabytes of data efficiently. For Google, the growth rate of the web and web technology changes were considered and also keep them up to date. Link structure and link text provide a lot of information for making a relevant judgement and quality filtering because of results relevant to searches needed to be returned. Google makes use of link structure.

Second most important feature for Google is PageRank. The citation (link) graph is an important resource for existing search engines. Larry Page and Sergey Brin created sitemaps as a sample that contained the hyperlinks. A *sitemap* is a file where web pages are listed on a website to tell Google and other search engines about the organization of website content. Search engine web crawlers like Google bot read this file to more intelligently crawl a website. Also, a sitemap can provide valuable *metadata* associated with the pages listed in that sitemap. Metadata is information about a webpage, such as when the page was last updated, how often the page is changed, and the importance of the page relative to other URLs on the site. These maps allowed rapid calculation of PageRank.

Citation literature has been applied to the web, largely by counting the backlinks to a given page. This gives some approximation of the importance of a page. It just does not count links only it takes the quality into account [1].

2.4 Web Mining

In the thesis paper of Xing et al. (2004), finding the content of the web and retrieving the user's interest and needs from their behaviour have been focused. Web mining is used for this purpose and it plays a very important role. HITS and PageRank algorithms are used for web structure mining. Both algorithms treat all the links equally when distributing rank scores. The Weighted PageRank algorithm, an extension to the standard PageRank algorithm is introduced in this paper.

The more popular webpages are, the more linkages other pages tend to have to them. Now the Weighted PageRank algorithm assigns larger rank values to more important (popular) pages than dividing rank values evenly among its outlink pages. Weighted PageRank takes into account the importance of both inlinks and outlinks of the pages. By considering these criteria a large number of relevant results are to a given query can be achieved compared to standard PageRank algorithm.

Web mining is the integration of information gathered by traditional data mining methodologies and techniques with information gathered over the World Wide Web. (*Mining* means extracting something useful or valuable from a baser substance, such as mining gold from the earth.) Web mining is used to understand customer behaviour, evaluate the effectiveness of a particular Web site, and help quantify the success of a marketing campaign. It makes utilization of automated apparatuses to reveal and extricate data from servers and web reports, and it permits organizations to get to both organized and unstructured information from browser activities, server logs, website and link structure, page content and different sources.

Web mining can be divided into three different types – **Web usage mining**, **Web content mining** and **Web structure mining**.

Web usage mining refers to the discovery of user access patterns from Web usage logs. **Web structure mining** tries to discover useful knowledge from the structure of hyperlinks. **Web content mining** aims to extract/mine useful information or knowledge from web page contents.

The goal of Web mining is to look for patterns in Web data by collecting and analyzing information in order to gain insight into trends, the industry and users in general [4].

2.5 Eigen Vector

In linear algebra, an eigenvector of a linear transformation is a non-zero vector whose direction does not change when that linear transformation is applied to it. More formally, if *T* is a linear transformation from a vector space *V* over a field *F* into itself and **v** is a vector in *V* that is not the zero vector, then **v** is an eigenvector of *T* if *T*(**v**)is a scalar multiple of **v**.

This condition can be written as the equation,

$$T(v) = \lambda v$$

Where λ is a scalar in the field F, known as the eigenvalue. A scalar is an element of a field which is used to define a vector space [6].

2.5.1 Power Iteration

In mathematics, the power method (also known as the power iteration) is an eigenvalue algorithm: given a matrix A, the matrix will produce a number λ which is the greatest eigenvalue of A and a nonzero vector v, the corresponding eigenvector of λ, such that $Av = \lambda v$. The power iteration is very simple algorithm but it may converge slowly. It can be used when A is a very large sparse matrix [7].

2.6 Types of Matrix

Matrixes can be two different types and the calculation of PageRank can be done by using both of them. First one is “**Sparse matrix**” and the second one is “**Dense Matrix**”. The idea of both is discussed below:

- **Sparse Matrix**: In computer science, a sparse matrix is a matrix in which most of the elements are zero. The number of zero-valued elements divided by the total number of elements (e.g., m × n for an m × n matrix) is called the sparsity of the matrix. For example, a **Stochastic Matrix** is a sparse matrix.

- **Dense Matrix**: In computer science, a dense matrix is a matrix in which most of the elements or all the elements are nonzero. For example, the **Google Matrix** is a dense matrix [8].

2.6.1 Advantages of Sparse Matrix

There are some advantages of a sparse matrix over a dense matrix. When storing and manipulating sparse matrices on a computer, it is beneficial and often necessary to use specialized algorithms and data structures that take advantage of the sparse structure of the matrix. Operations using standard dense-matrix structures and algorithms are slow and inefficient when applied to large sparse matrices as processing and memory are wasted on the zeroes. Sparse data is by nature more easily compressed and thus require significantly less storage. Some very large sparse matrices are infeasible to manipulate using standard dense-matrix algorithms.

2.7 Off-Page SEO

Off-page SEO refers to activities someone can perform outside the boundaries of a website. The most important are:

1. Link Building
2. Social Media Marketing
3. Social Bookmarking

2.7.1 Link Building

Link building is the most popular and **effective** Off-Page SEO method. Basically by building external links to a website, someone is trying to gather as many ‘votes’ as possible to bypass competitors and rank higher.

For example, consider two website A and B. If website A has its link on website B, then for website A it will be “Link Building”. And for website B it will be a “Backlink”.

Over the years webmasters have been trying to build links to their websites to get higher rankings and they ‘invented’ a number of ways to increase link count. The most popular ways were:

- **Blog Directories** – something like yellow pages but each entry had a link pointing to a website.

  Yellow page refers to a telephone directory of businesses, organized by category rather than alphabetically by business name, and in which advertising is sold.

- **Blogging** - Blogging is one of the best ways to promote a website online. By writing a blog for a website, it gives a reason for visitors to keep returning to the site and keep up to date with all the latest posts. It also helps search engines to crawl the site more frequently, as they have to update those latest blog post entries, which ultimately helps rank higher in search engine results pages (SERPs).

  - **Procedures of posting in blog sites**
    - Complete Registration
    - Choose a topic
    - Give title
    - Write description
    - Add picture
    - Add label so that people can find your post with those words

After following all the above procedures, post can be published.

- **Forum Signatures** – Many people were commenting on forums for the sole purpose of getting a link back to their website (they included the links in their signature).

- **Comment link** – The same concept as forum signatures where comment can be made on some other website or blog in order to get a link back. Even worse, instead of using real name keywords can be used for writing. For example, ‘comment by Alex Chris’, can be written as ‘comment by How to lose weight’.

- **Article Directories** – By publishing articles in article directories links can be collected back to website. Some article directories accepted only unique content while other directories accepted anything from spin articles to already published articles.

  Article directories allow users to submit unique articles to the directory for content syndication. These directories allow articles to embed links to other websites with relevant anchor text. Popular article directories are considered authority sites and are constantly crawled by search engine bots. Webmasters submit articles with relevant anchor text linking back to their site to obtain backlinks.

- **Search Engine Submission -** Search engines will eventually find a site online, but that can take a while. To speed everything up, websites should be submitted to the most popular search engines like Google, Yahoo, Bing, etc.

- **Video Marketing –** If any videos are used on a site, then it can be submitted to sites like; YouTube. This will allow people to find contents in other ways.

- **Shared Content Directories** – Websites like hubpages and infobarrel allows to publish content and in return couple of links pointing to websites.

- **Link exchange schemes** – Instead of trying to publish content randomly, links could be exchanged with other webmasters.

2.7.2 The birth of “Black hat SEO”

Link building was an easy way to manipulate the search engine algorithms and many spammers tried to take advantage of this by building link networks which gradually lead to the creation of what is generally known as black hat SEO.

Google has become very intelligent in recognizing black hat techniques and with the introduction of Panda, Penguin and Hummingbird (that’s how the Google Algorithm releases are called), they have managed to solve the problem and protect their search engine results from spammers.

Of course there are still exceptions but they are doing advances in every new algorithmic release and soon enough none of these tricks will work.

2.7.3 Social Media

Social media is part of ‘Off-Page SEO’ and it’s also a form of link building. The likes of Facebook, Twitter, Google+ plays an important role for link building.

Social Media mentions are gaining ground as ranking factors and proper configuration of social media profiles can also boost SEO.

2.7.4 Social Bookmarking

Social bookmarking is not as popular as it used to be in the past but it is still a good way to get traffic to websites. Depending niche, websites like reddit.com, stumbleupon.com, scoop.it and delicious.com (to name a few) to promote contents.

2.7.5 Importance of Off-Page SEO

Search engines have been trying for decades to find a way to return the best results to the searcher.

To achieve this, some of the Off-Page SEO techniques are needed to be taken into account.

Off page SEO gives a very good indication on how the World (other websites and users) perceive a particular website. A web site that is high quality and useful is more likely to have references (links) from other websites; it is more likely to have mentions on social media (Facebook likes, tweets etc.) and it is more likely to be bookmarked and shared among communities of like-minded users.

2.7.6 Benefits of Off-Page SEO to a Website

A successful off-site SEO strategy will generate the following benefits to website:

- **Increase in rankings** – The website will rank higher in the SERPs and this also means more traffic.

  A **search engine results page** (**SERP**) is the page displayed by a search engine in response to a query by a searcher. The main component of the SERP is the listing of results that are returned by the search engine in response to a keyword query, although the page may also contain other results such as advertisements.

  The results are of two general types, organic (i.e., retrieved by the search engine's algorithm) and sponsored (i.e., advertisements). The results are normally ranked by relevance to the query. Each result displayed on the SERP normally includes a title, a link that points to the actual page on the Web and a short description showing where the keywords have matched content within the page for organic results. For sponsored results, the advertiser chooses what to display.

- **Increase in PageRank** – Page rank is a number between 0 and 10 which indicates the importance of a website in the eyes of Google. It is the system invented by Larry Page and Sergey Brin (Google founders) and one of the reasons that Google was so successful in showing the most relevant results to the searcher. Page rank today is only one out of the 200 factors that Google is using to rank websites.

More exposure – Higher rankings also means greater exposure because when a website ranks in the top positions: it gets more links, more visits and more social media mentions. It’s like a never ending sequence of events where one thing leads to another and then to another etc. [9].

2.8 On-Page SEO

1. **Keyword in the title tag -** The title meta tag is one of the strongest relevancy signals for a search engine. The tag itself is meant to give the accurate description of the pages content. Search engines use it to display the main title of a search result. Including a keyword in it will indicate to search engine how to rank the page.
   Ideally, the keyword should be placed at the start of the title tag. Pages optimized this way will rank better than those with keyword closer to the title's tag end.

2. **Keyword in meta description tag -** The importance of the meta description tag today is often discussed in SEO circles. It is nonetheless still a relevancy signal. It is also crucial for gaining user clicks from search results pages. Including the keyword in it makes it more relevant to a search engine and a searcher

3. **Keyword and Title in H1 tag -** H1 tag is yet another relevance factor, serving as a description of the pages content. In spite of an ongoing discussion about its importance, it is still a good practice to include your keyword in a unique H1 tag on a page. The H1 tag is the headline tag. WordPress automatically adds title in the H1 tag.

4. **The length of the content -** These days searchers want to be educated and won't satisfy with basic information. Google, therefore, looks for authoritative and informative content to rank first. And its' common sense that the longer your content is, the greater the chance that you can cover more aspects of your topic. Don't be shy of writing long but highly useful copy then.

5. **Image Optimization -** It's not only text that can be optimized on a page but other media too. Images, for instance, can send the search engine relevancy signals through their alt text, caption, and description for example.

6. **Dropping Keywords in first 100 words –** If the keywords drop inside the first 100 words, it is better.

7. **Responsive Design –** Specially mobile friendly website design helps very much for SEO

8. **Using the social sharing buttons –** Social shares generate more eyeballs on the content of the website. Social media like Facebook, Twitter, Google+, LinkedIn plays very important part.

9. **Outbound links -** Linking to authoritative pages sends trust signals to the search engine. This can be a huge trust factor for Google. Too many outbound links, however, can significantly diminish the page's PageRank, hurting its search visibility. Outbound links can affect your rankings but use them in moderation.

10. **Internal links -** Interlinking pages on website can pass their strength between them.

11. **Keyword in URL -** Including the keyword in the URL slug (that's the bit that appears after the ".com/"part of the URL) is said to send another relevancy signal to Google [10].

Table 2.1: Criteria of On-Page and Off-Page SEO

Number of Criteria	Off-Page SEO	On-Page SEO
1	Link Building	Keyword in the title tag
2	Social Media Marketing	Keyword in meta description tag
3	Social Bookmarking	Keyword and title in H1 tag
4		The length of the content
5		Image Optimization
6		Dropping keywords in first 100 words
7		Responsive Design
8		Using the social sharing buttons
9		Outbound links
10		Internal links
11		Keyword in URL

2.9 PageRank: The Complete Algorithm

- Input: Graph G and parameter β
    - Directed graph G with spider traps and dead ends
    - Parameter β

```
➢ Output: PageRank vector r
    • Set: $r_j^{(0)}=\frac{1}{N}$ , t=1
    • Do:
        ▪ $\forall j: r'^{(t)}_j = \sum_{i\to j}\beta\frac{r_i^{(t-1)}}{d_i}$
          $r'^{(t)}_j$=0 if in degree of j is 0
        ▪ Now re-insert the leaked PageRank:
          $\forall j: r_j^{(t)}=r'^{(t)}_j + \frac{1-S}{N}$where: S=$\sum_j r'^{(t)}_j$
        ▪ $t = t + 1$
    • While $\sum_j \left|r_j^{(t)} - r_j^{(t-1)}\right|>\varepsilon$ [11]
```

2.10 Sample Web Graph

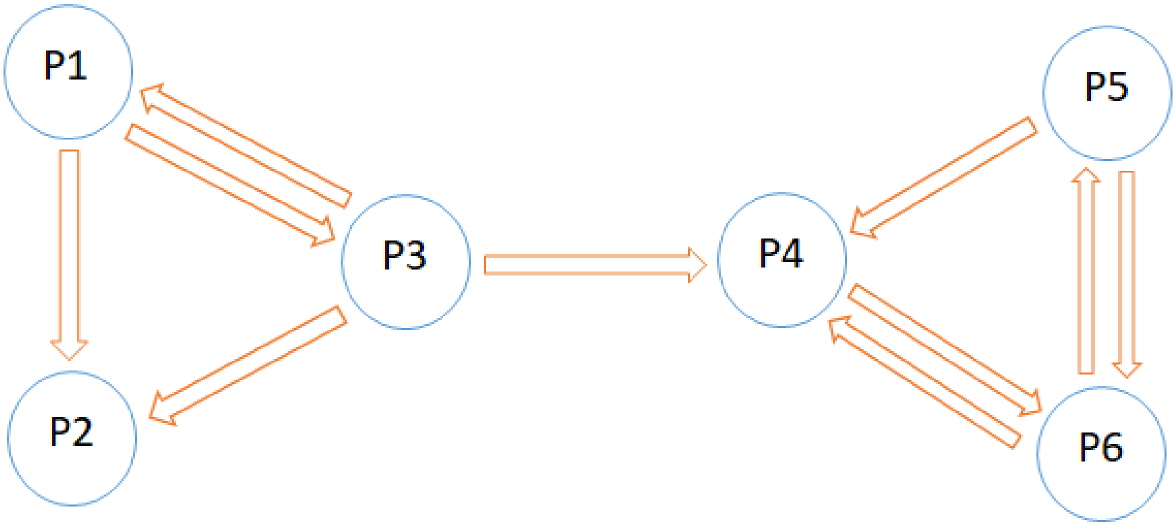


Figure 2.1: A Sample Web Graph

The system of web pages with hyperlinks between them is viewed as a directed graph called a hyperlink graph of the system. The nodes are the pages and there is an edge from page P_i to page P_j if there is a hyperlink that points from P_i to page P_j; this is an out-link from page P_i and an in-link to P_j.

The figure above is a hyperlink graph of a system of six web pages. We can see, for example, that there are out-links from page P_3 to pages, $P_{1,}$ P_2 and P_4. We can also see that P_2 has no outlinks at all.

2.10.1 PageRank Equation of Sample Web Graph

1. PR (P1) =$\frac{PR(P3)}{3}$
2. PR (P2) =$\frac{PR(P1)}{2} + \frac{PR(P3)}{3}$
3. PR (P3) =$\frac{PR(P1)}{2}$
4. PR (P4) =$\frac{PR(P3)}{3} + \frac{PR(P5)}{2} + \frac{PR(P6)}{2}$
5. PR (P5) =$\frac{PR(P6)}{2}$
6. PR (P6) =$\frac{PR(P4)}{1} + \frac{PR(P5)}{2}$

From the above graph, we can see that page P_1 has got one in link from only page P_3. So, the *PageRank* of the page P_1 will be equal to the *PageRank* of page P_3divided by the total number of out-links page P_3 has got, which is 3. Thus,

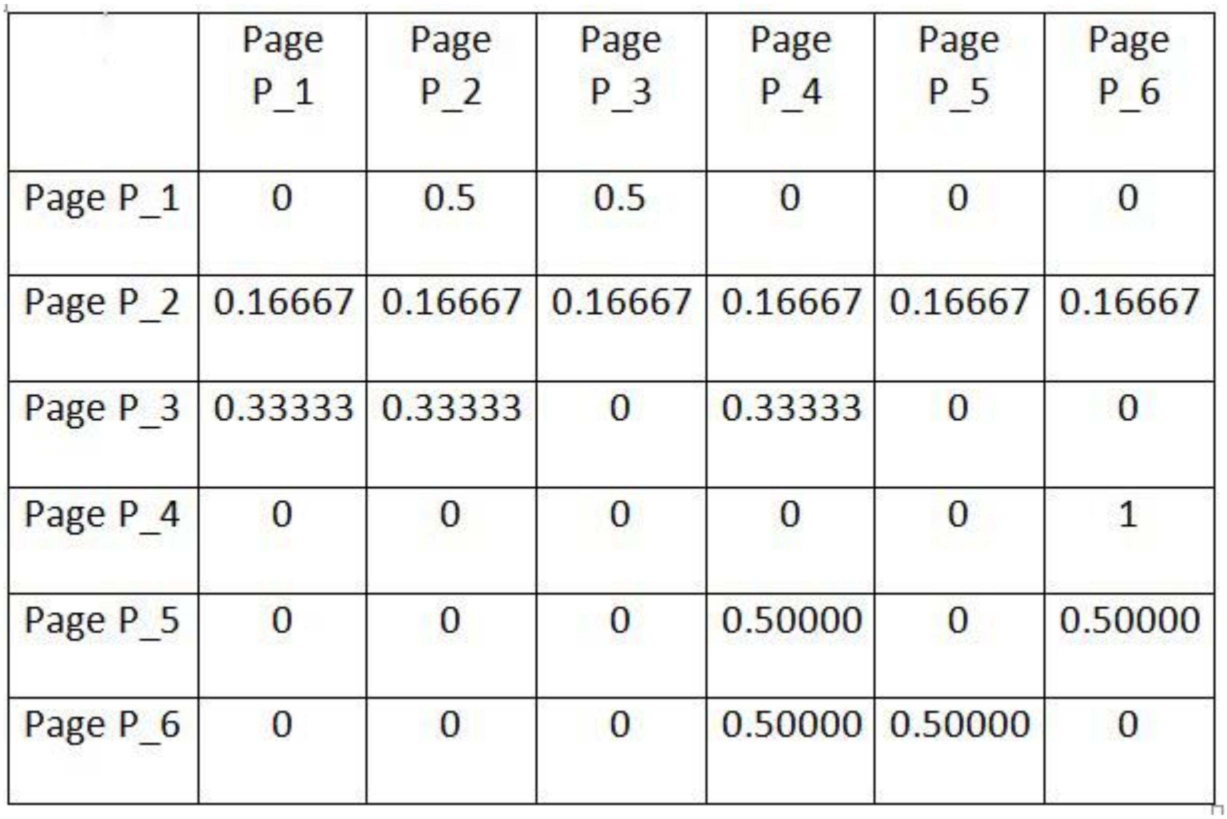
PR (P1) =$\frac{PR(P3)}{3}$

Same goes for all the other pages.

2.11 The Stochastic Matrix

In mathematics, a **stochastic matrix** is a square matrix used to describe the transitions of a Markov chain. Each of its entries is a nonnegative real number representing a probability. It is also known as **probability matrix** or **Markov matrix** [12].

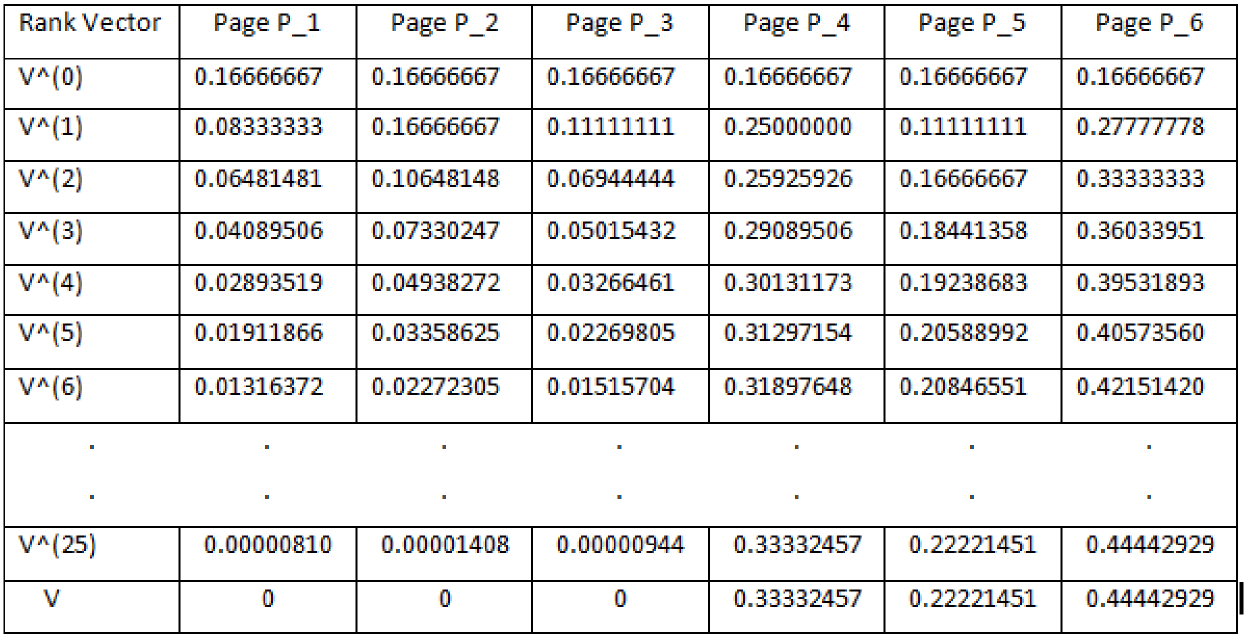
Table 2.2: Stochastic Matrix ‘S’



	Page P_1	Page P_2	Page P_3	Page P_4	Page P_5	Page P_6
Page P_1	0	0.5	0.5	0	0	0
Page P_2	0.16667	0.16667	0.16667	0.16667	0.16667	0.16667
Page P_3	0.33333	0.33333	0	0.33333	0	0
Page P_4	0	0	0	0	0	1
Page P_5	0	0	0	0.50000	0	0.50000
Page P_6	0	0	0	0.50000	0.50000	0

We are using the stochastic matrix to solve the dangling node problem. The dangling node problem means that a node in our web graph has no outlink. If a web surfer reaches such node, he will be stuck there. So, when a web surfer encounters such problem, there is a probability that he will leave the web page. Then he might randomly select another web page and start surfing again. This is how the stochastic matrix solves the dangling node problem.

Table 2.3: Result after 25 Iterations Using Stochastic Matrix



Rank Vector	Page P_1	Page P_2	Page P_3	Page P_4	Page P_5	Page P_6
V^(0)	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667
V^(1)	0.08333333	0.16666667	0.11111111	0.25000000	0.11111111	0.27777778
V^(2)	0.06481481	0.10648148	0.06944444	0.25925926	0.16666667	0.33333333
V^(3)	0.04089506	0.07330247	0.05015432	0.29089506	0.18441358	0.36033951
V^(4)	0.02893519	0.04938272	0.03266461	0.30131173	0.19238683	0.39531893
V^(5)	0.01911866	0.03358625	0.02269805	0.31297154	0.20588992	0.40573560
V^(6)	0.01316372	0.02272305	0.01515704	0.31897648	0.20846551	0.42151420
.	.	.	.	.	.	.
.	.	.	.	.	.	.
V^(25)	0.00000810	0.00001408	0.00000944	0.33332457	0.22221451	0.44442929
V	0	0	0	0.33332457	0.22221451	0.44442929

But, there is a problem in the Stochastic Matrix. After 25 iterations, we can see that our web pages converge and the summation of the vector is 1. So we can think that we have reached our goal. But the problem is that page P_1, P_2, P_3 has no page rank value. So this is a major problem because every web page has importance whether it is too big or too small.

To solve the problem Google uses the Google matrix G.

The formula for Google matrix is,

$$dS + (1\text{-}d)(\frac{1}{N})_{N*N}$$

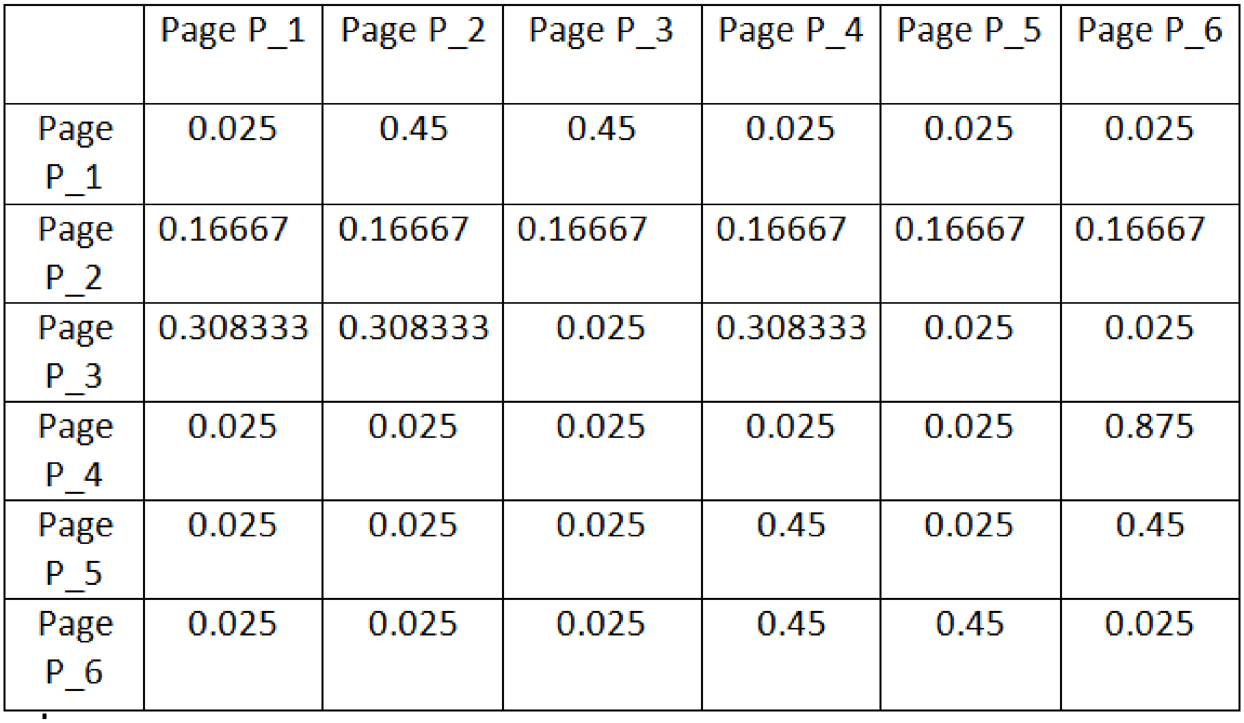
Here,

S is the Stochastic Matrix and d is the damping factor.

2.12 The Google Matrix

The Google matrix **“**G**”** of a directed network is a stochastic square matrix with non-negative matrix elements and the sum of elements in each column being equal to unity. This matrix describes a Markov matrix of transitions of a random surfer performing jumps on a network of nodes connected by directed links [12].

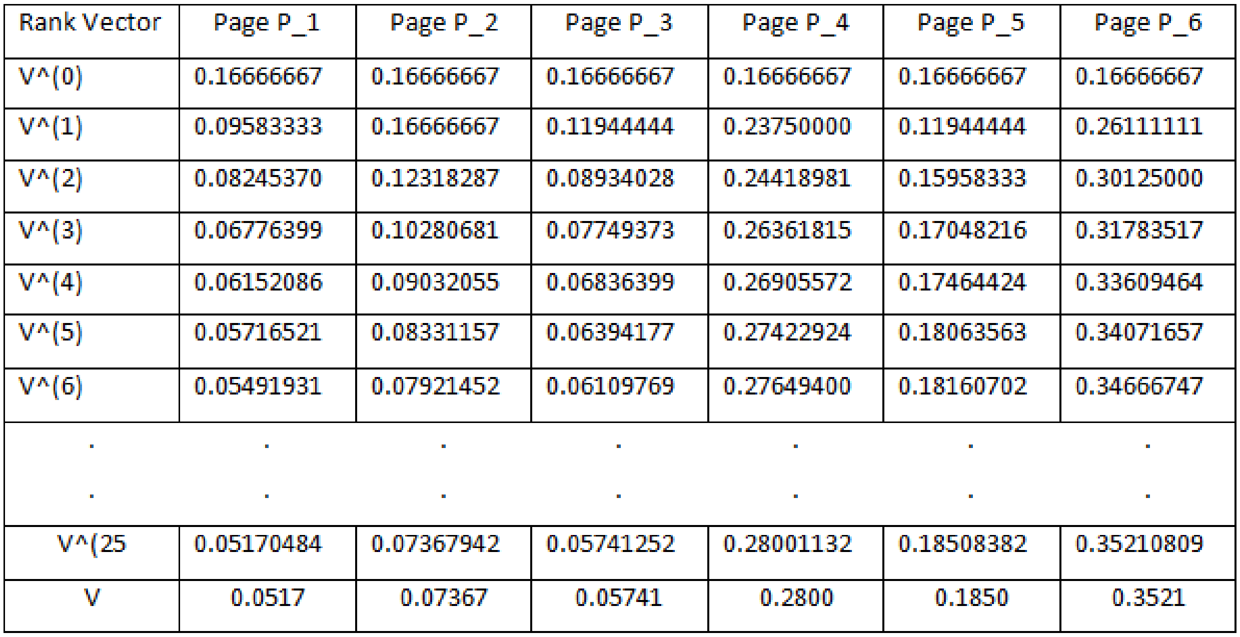
Table 2.4: Google Matrix ‘G’



	Page P_1	Page P_2	Page P_3	Page P_4	Page P_5	Page P_6
Page P_1	0.025	0.45	0.45	0.025	0.025	0.025
Page P_2	0.16667	0.16667	0.16667	0.16667	0.16667	0.16667
Page P_3	0.308333	0.308333	0.025	0.308333	0.025	0.025
Page P_4	0.025	0.025	0.025	0.025	0.025	0.875
Page P_5	0.025	0.025	0.025	0.45	0.025	0.45
Page P_6	0.025	0.025	0.025	0.45	0.45	0.025

This is the Google Matrix of our sample web graph. Here in this matrix, we can observe every row and column has a nonzero element.

Table 2.5: Result after 25 Iterations Using Google Matrix



Rank Vector	Page P_1	Page P_2	Page P_3	Page P_4	Page P_5	Page P_6
V^(0)	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667
V^(1)	0.09583333	0.16666667	0.11944444	0.23750000	0.11944444	0.26111111
V^(2)	0.08245370	0.12318287	0.08934028	0.24418981	0.15958333	0.30125000
V^(3)	0.06776399	0.10280681	0.07749373	0.26361815	0.17048216	0.31783517
V^(4)	0.06152086	0.09032055	0.06836399	0.26905572	0.17464424	0.33609464
V^(5)	0.05716521	0.08331157	0.06394177	0.27422924	0.18063563	0.34071657
V^(6)	0.05491931	0.07921452	0.06109769	0.27649400	0.18160702	0.34666747
. .	. .	. .	. .	. .	. .	. .
V^(25	0.05170484	0.07367942	0.05741252	0.28001132	0.18508382	0.35210809
V	0.0517	0.07367	0.05741	0.2800	0.1850	0.3521

Now we can rank all the web pages mentioned in the sample web graph. The ranking will be,

- Rank 1: Page ⇨P6
- Rank 2: Page ⇨P4
- Rank 3: Page ⇨P5
- Rank 4: Page ⇨P2
- Rank 5: Page ⇨P3
- Rank 6: Page ⇨P1

2.12.1 Limitation of Google Matrix

The Google Matrix successfully ranks all the web pages which seem to be very efficient. But the problem with Google Matrix is that the memory space. The World Wide Web is huge. It is not possible to store the result of such huge matrix in any memory after every iteration.

2.12.2 Solution of Google Matrix

Say, $r^{new} = A.\ r^{old}$; where A = Google Matrix and r = rank vector

Here $A_{ij} = \beta.M_{ij} + (1-\beta)/N$; where M = Stochastic Matrx

Now, $r_i = \sum^{N}_{j=1} A_{ij}\ .\ r_j$

$= \sum^{N}_{j=1} [\ \beta.M_{ij} + (1-\beta)/N\ ]\ .\ r_j$

$= \sum^{N}_{j=1} \beta.M_{ij}\ .\ r_j + \sum^{N}_{j=1} (1-\beta)/N\ .\ r_j$

$= \sum^{N}_{j=1} \beta.M_{ij}\ .\ r_j + (1-\beta)/N$; as $\sum^{N}_{j=1} r_j = 1$

So, we get $r = \sum^{N}_{j=1} \beta.M_{ij}\ .\ r_j + (1-\beta)/N$

The Google Matrix had the memory size problem. So, to solve the problem a new matrix is introduced where it uses the sparse matrix and not the dense matrix anymore.

2.13 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

2.14 CSS

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS stands for **Cascading Style Sheets**.

It styles the HTML elements and describes how they will be displayed. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure from presentation.CSS saves a lot of works and time. The webpages load faster for the use of it as it significantly reduces the file transfer size.

2.15 K-Means Clustering

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both k-means and Gaussian Mixture Modeling. Additionally, they both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.

The algorithm has a loose relationship to the k-nearest neighbor classifier, a popular machine learning technique for classification that is often confused with k-means because of the k in the name. One can apply the 1-nearest neighbor classifier on the cluster centers obtained by *k*-means to classify new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm [13].

2.16 Scikit-Learn

Scikit-learn (formerly **scikits.learn**) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy [14].

Clustering of unlabeled data can be performed with the module sklearn.cluster.

Each clustering algorithm comes in two variants: a class that implements the **fit** method to learn the clusters on train data, and a function, that, given train data, returns an array of integer labels corresponding to the different clusters. For the class, the labels over the training data can be found in the **labels_** attribute.

Plenty of models have fit methods in scikit-learn. When you call fit method it estimates the best representative function for the data points (could be a line, polynomial or discrete borders around). With that representation, you can calculate new data points. Take linear regression for example: when you call fit on a dataset of points, it'll give you a function that represents a line that is **best fits** all the points. With that line function you can estimate other results [15].

Chapter 3

Proposed Model

For our model, we have used two websites. One website has all the Search Engine Optimization (SEO) factors and the other one only contains few SEO factors. First website with the SEO factors is http://trainingwithvick.com/ and the one with few SEO factors is trainingalongvick.000webhostapp.com/ . We will apply the Search Engine Optimization (SEO) criteria to both websites.

The websites that we are using are basically related to digital information business. A digital information business website is such website that sells digital products. By using the word information it means that it is not just some simple information rather than a guidance which makes our life better.

Both of our experimental website serially, http:\\trainingwithvick.com and trainingalongvick.000webhostapp.com comes up with the idea of selling digital information to the users. Anybody can have a small business online where they can sell their product and earn profit quickly. These two website has this motto to help them as much as possible. The website home page is describes the website and its purpose and the rest of the website pages are advertisements that has been posted.

The benefit is also a matter of concern as having a website to help others has to have some reason behind it. So by promoting other people or companies product and getting some profit out of it is known as “Affiliate marketing”.

Other reason for selling digital information online is that, they are easy and inexpensive to create. Only some good idea is needed. Some notable digital information can be

1. Graphics
2. Software
3. E-book
4. Website themes
5. Online course
6. Tutorials (Audio and Video)

3.1 Technologies Used

These technologies are used for developing the website:

- HTML
- CSS

3.2 Word2Vec

We will use Word2vec for our keyword research for On-page SEO criteria. Word2Vec is an estimated representation of word in a vector space model. The vector representation of a word is called “Word Embedding”. It gives an idea how the words are closely related to each other in some given domains. Similar words which are semantically similar are mapped in a nearby point. This algorithm was created by a team of researchers led by Tomas Mikolov at Google.

3.2.1 Procedure of Word2vec

Word2Vec can be implemented in languages like C, Java, Python etc. It takes a whole sentence or sting as input then breaks them up a chunk of words that appear in the sentences. Lastly it creates a vector representation of those words. There is a very efficient library found in python language that works very well to use Word2Vec. The library is called “Gensim”.

3.2.2 Useful Features of Gensim

- **Similarity Queries**: Gensim contains code for fast indexing of documents in their semantic representation and fast retrieval of topically similar documents. Which will be a key factor in our work.

- **Platform Independent**: As it is pure python, it runs on Linux, Windows and OS X.

3.2.3 Word2Vec Steps

- The keywords will be first analyzed from Google Keyword Planner. The keywords will be stored.

- Then we will collect site texts. Relevant data from different websites across the web. Data will be collected in the form of sentences.

- Then we will use Word2Vec to make chunk of words from the sentences. After that the vector representation model will be created.

- Then we used K-means clustering to cluster the words that we found from the sentences. We used Scikit-learn for Word2Vec to create clusters. Then we plotted those clusters and we tried to which words are nearest to their respective centroid. Then we will take those words and compared with our collected keywords. Then we will use those words in our title and meta-description as well as in our website body.

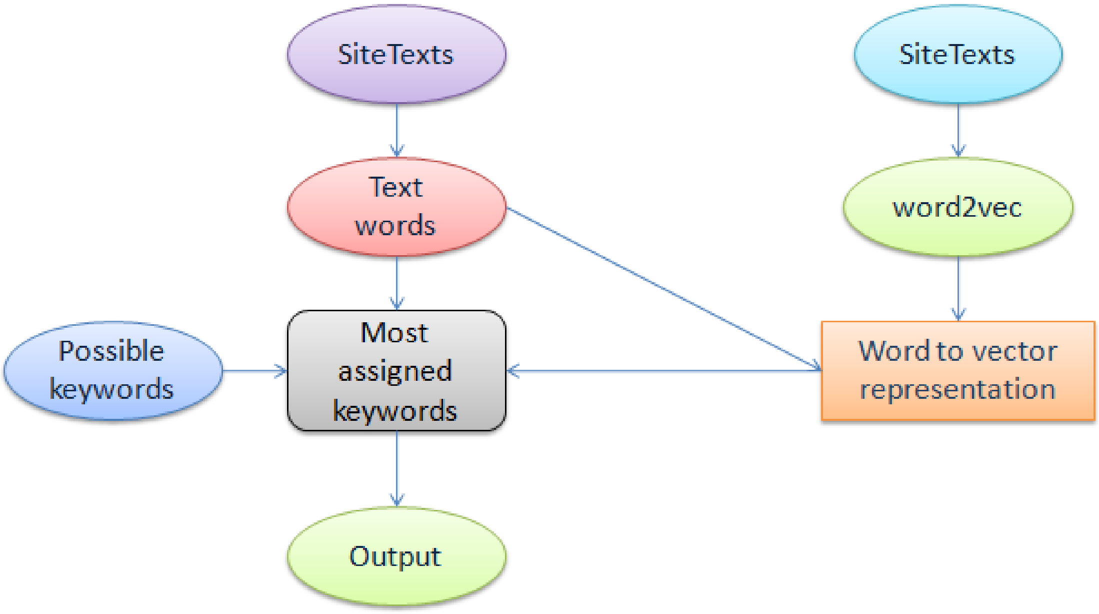


Figure 3.1: Flow chart for Word2Vec

At the very beginning, collecting the site or websites texts was a challenging task. Because it is very much important to find the relevant texts from the website which were related to our experimental websites. As both of our websites had similar features, our collection of site texts helped for both of them. Some of the most notable websites in the field of digital information selling were Shopify, E-junkie, ClickBank and many more. We got most of our site texts from these mentioned websites.

We made sure that the collected site texts had those keywords we collected specially from Keyword Planner. This made our work relatively easier to implement. From there we could really compare that which collected words or new words are important candidate to become the keywords.

As we were working on python, various models made our work easy. The most amazing part was that we could visualize the results in front of us as well as we could see them individually. We also had the independence of training our samples from time to time so that we can a better approximation.

We were successful on this method for finding some keywords out of plain site text apart from selected keywords and used them on our experimental website with our convenience.

Chapter 4

Simulation

4.1 SEO Procedure

As it has been mentioned in previous chapter that we can apply SEO technique in a website in two ways, Off-page SEO and On-page SEO. Now, we have applied the on page criteria on both websites that we have created.

4.1.1 On-Page Procedure

On-Page SEO means how we have created our website that contained all the factors in order to be search friendly. This can be done by using **Keywords** in the website.

We found the right keywords for our website by using the **Keyword Planner** of **Google AdWords**. While we were researching for the keywords, we had to keep in mind that we would only use keywords with medium or low competition. Because if the keyword has lower competition, our website will have a better chance to be in the top list of a search result when people type queries in Google to search for the exact thing they need. So, we were careful for choosing our keywords.

After finding the right keyword, we have used them in our website's meta-description, title tag and other descriptive areas. The **meta-description** is a ~160 character snippet which is a tag in HTML that summarizes a page's content. Search engines show the meta-description in search results mostly when the searched for phrase is contained in the description.

We implemented internal link in our website. The internal link helps our websites pages to stay connected internally.

4.1.2 Off-Page Procedure

Off-Page SEO completely depends on marketing and advertising our website. Off-Page SEO is a time consuming procedure. It takes time to get enough inbound links in order to have a high PageRank value. It is known that to achieve the PageRank value 1 from 0, we will need roughly 100 clicks on our website's URL link everyday. The highest PageRanked websites get so many clicks each day and the usage of such websites are very frequent.

4.2 Implementing Word2Vec

We implemented Word2Vec in python language with the help of Gensim library. The procedure that was proposed in the last chapter, we implemented some part of it. We collected texts from sites, then created the chunk of words. After that, we created their vector representation model. Here is a small representation of it.

- At first, we took some sentences. Among them two sample sentences are:
  1. "For more than 20 years, ClickBank has delivered lifestyle products to customers around the globe."
  2. "Each of our unique products is created by a passionate entrepreneur focused on improving the lives of our customers by inspiring, instructing, or coaching."

- Then we made the chunk of data from those sentences. The Gensim library took every word from the sentences and created individual words.

```
{'For': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9391DD8>,
'more': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9391B00>,
'than': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9391C50>,
'20': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93915C0>,
'years': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9391908>,
'Clickbank': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93912E8>,
'has': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9391C88>,
'delivered': <gensim.models.keyedvectors.Vocab object at 0x0000003DF7C14D30>,
'lifestyle': <gensim.models.keyedvectors.Vocab object at 0x0000003DF7C143C8>,
'products': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93725F8>,
'to': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9372F60>,
'customers': <gensim.models.keyedvectors.Vocab object at 0x0000003DF94EA278>,
'around': <gensim.models.keyedvectors.Vocab object at 0x0000003DF94B59B0>,
'the': <gensim.models.keyedvectors.Vocab object at 0x0000003DF94B5550>,
'globe': <gensim.models.keyedvectors.Vocab object at 0x0000003DF94B5A90>,
'Each': <gensim.models.keyedvectors.Vocab object at 0x0000003DF94B55C0>,
'of': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9443DA0>,
'our': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9436898>,
'unique': <gensim.models.keyedvectors.Vocab object at 0x0000003DF9423278>,
'is': <gensim.models.keyedvectors.Vocab object at 0x0000003DF935F6A0>,
'created': <gensim.models.keyedvectors.Vocab object at 0x0000003DF935FF28>,
'by': <gensim.models.keyedvectors.Vocab object at 0x0000003DF935FB70>,
'a': <gensim.models.keyedvectors.Vocab object at 0x0000003DF935FB00>,
'passionate': <gensim.models.keyedvectors.Vocab object at
0x0000003DF935FEF0>,
'entrepreneur': <gensim.models.keyedvectors.Vocab object at
0x0000003DF7C04E48>,
'focused': <gensim.models.keyedvectors.Vocab object at 0x0000003DF7C04A58>,
'on': <gensim.models.keyedvectors.Vocab object at 0x0000003DF7C04F98>,
'improving': <gensim.models.keyedvectors.Vocab object at 0x0000003DF7C041D0>,
'lives': <gensim.models.keyedvectors.Vocab object at 0x0000003DF7C04080>,
'inspiring': <gensim.models.keyedvectors.Vocab object at 0x0000003DF7C04940>,
'instructing': <gensim.models.keyedvectors.Vocab object at
0x0000003DF7C045F8>,
'or': <gensim.models.keyedvectors.Vocab object at 0x0000003DF94C8EB8>,
'coaching': <gensim.models.keyedvectors.Vocab object at 0x0000003DF938EEF0>,
'As': <gensim.models.keyedvectors.Vocab object at 0x0000003DF938EF28>, 'one':
<gensim.models.keyedvectors.Vocab object at 0x0000003DF938E320>,
'largest': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93BAF98>,
'online': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93BA630>,
'retailers': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93BAE48>,
'ClickBank': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93BA198>,
'harnesses': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93BA9B0>,
'awesome': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93BAE80>,
'power': <gensim.models.keyedvectors.Vocab object at 0x0000003DF93BAC88>,
'digital': <gensim.models.keyedvectors.Vocab object at 0x0000003DFA473780>,
'marketing': <gensim.models.keyedvectors.Vocab object at 0x0000003DFA473BE0>,
'partners': <gensim.models.keyedvectors.Vocab object at 0x0000003DFA473860>,
'combinedwith': <gensim.models.keyedvectors.Vocab object at
0x0000003DFA4800F0>,
```

Figure 4.1: Individual words from each sentence

- After we made the sentences in their individual word representation, we are now ready to make their vector representation. As for simplicity we are presenting the vector representation of a single word.

```
  Products
    [-2.3762004e-03  3.1779357e-04  3.3622305e-03  7.0507522e-04
    1.9933961e-03 -2.7150752e-03  3.3935517e-04  2.3237516e-03
   -4.6076155e-03  2.3197639e-03 -2.0174359e-03 -4.3869563e-03
   -4.8714927e-03  1.2556434e-03 -1.8354512e-03  2.6071109e-03
    2.6171959e-03  3.6018265e-03 -3.1439427e-03 -4.3593366e-03
   -6.2448620e-05  4.2396360e-03 -2.4743092e-03 -1.2193115e-03
   -4.3518841e-03 -1.9727589e-04  1.1769851e-03 -3.0460318e-03
    1.9603749e-03 -1.1068626e-03 -2.9955872e-03 -8.7402365e-04
    3.8393694e-03  7.4668060e-04  3.3889036e-03  2.7345605e-03
   -7.6061208e-04 -3.5967536e-03  6.6690729e-04 -2.7349233e-03
   -1.5938244e-03 -1.9151851e-03 -4.4448171e-03 -2.7891991e-03
    4.4670724e-03  2.3192470e-03 -1.5241043e-03  4.8036152e-03
    8.0101978e-04  5.4202962e-04  4.7600429e-04  2.7451437e-04
    1.1131940e-03  2.2646512e-03 -3.0126753e-03  3.7672964e-03
   -8.3568296e-04  1.6293188e-03 -1.9665945e-03  6.9731812e-04
    3.8827474e-03 -8.0281228e-05  5.2768868e-05  3.3269918e-03
   -4.1958317e-03  3.3391085e-03 -3.2802371e-03 -4.8928517e-03
   -2.6326659e-03 -2.7017905e-03  4.0609953e-03  9.7919651e-04
    1.0801597e-03  2.6256123e-03  3.2536201e-03  2.1599585e-04
   -2.5132291e-03 -2.4392817e-03 -2.5960186e-03 -1.3657066e-03
   -8.1702863e-04  3.5618659e-04 -7.0506247e-04  2.0909128e-03
   -1.8718581e-03 -4.2748032e-03  2.0666029e-03 -2.1174124e-03
    3.2930665e-03  4.8447247e-03  3.7416103e-03  2.4765676e-03
   -3.7897041e-03  3.6836611e-03 -4.5489534e-04  4.2906553e-03
   -1.6244800e-03  1.5096893e-04 -2.0606362e-03  3.2459267 e-03]
```

Figure 4.2: Vector representation of the word "Products"

- Using K-means clustering algorithm, we created clusters of our words.

```
Cluster id labels for inputted data
[3 6 1 1 6 0 3 5 3 6 5 2 5 7 2 0 5 6 0 2 2 2 8 4 3 8 8 4 7 2 1 2 7 2 5 3 0
 0 3 0 4 7 2 2 2 8 1 3 3 8 4 0 3 2 8 4 9 6 3 5 4 9 4 7 4 2 9 5 2 0 6 8 1 0
 1 5 2]
```

Figure 4.3: Labels of input words

- Then we also found out the centroids of our input words.

```
Centroids data
   [ 2.57375976e-03  1.16536859e-03 -1.28467311e-03  3.24269582e-04
   1.88396894e-04  4.68635117e-05 -6.55569835e-04  5.33394923e-04
   8.37678439e-04  1.09878660e-03  6.62226055e-04  1.56753557e-03
   9.87748499e-04  5.09300153e-05  1.90996667e-04 -1.08176922e-04
  -2.11337203e-04  3.99946497e-04 -2.49624625e-03 -2.20593228e-03
   7.90734601e-04 -1.65867968e-04 -5.17182110e-04 -1.37606519e-04
   1.14954868e-03  1.90943643e-03  7.69273494e-04  1.74097670e-03
   1.32865767e-04  3.26683657e-04  4.09673259e-04 -1.03377842e-03
  -1.43570511e-03 -7.97215151e-04  9.55676602e-04  2.82814773e-03
  -2.25834223e-03  1.94085354e-04  1.07211526e-03 -1.94807490e-03
   2.78481952e-04  5.37942396e-05  9.86170722e-04  7.39888113e-04
  -8.47485062e-05  1.39367569e-03  1.80258031e-03  1.49183010e-03
  -2.66880204e-04  3.46370391e-04 -3.41344683e-04  8.50429758e-04
   9.09672526e-04 -8.92684388e-04 -8.47386080e-04 -3.14654462e-04
   1.98663212e-03  8.04090756e-04 -4.81122639e-04 -1.21454103e-03
   1.75925368e-03  1.30392658e-03  2.25168216e-04 -8.72851000e-04
   1.35124731e-03 -7.18359312e-04  1.13555312e-03  1.72358670e-03
   1.54855865e-04  1.08809187e-03 -9.67767031e-04 -8.48610303e-04
   2.25448486e-04 -1.20182987e-03  9.17116355e-04 -1.81244360e-03
  -2.53276550e-03 -1.34194398e-03  6.41674851e-05  5.37794607e-04
  -1.84779195e-03 -1.90160458e-03 -1.53438526e-03  1.69204175e-03
  -4.72390093e-05  9.08473012e-05 -8.42355366e-04 -1.01700462e-04
   5.57176594e-04  1.85549099e-04  2.23982497e-04  1.15996483e-03
   7.58387032e-04 -2.14195391e-03  1.17946649e-04  8.86039110e-04
   6.16058765e-04 -9.10496979e-04 -2.54798477e-04  9.31426883e-04]
```

Figure 4.4: Sample Centroid Data

- After that, we plotted our clusters.

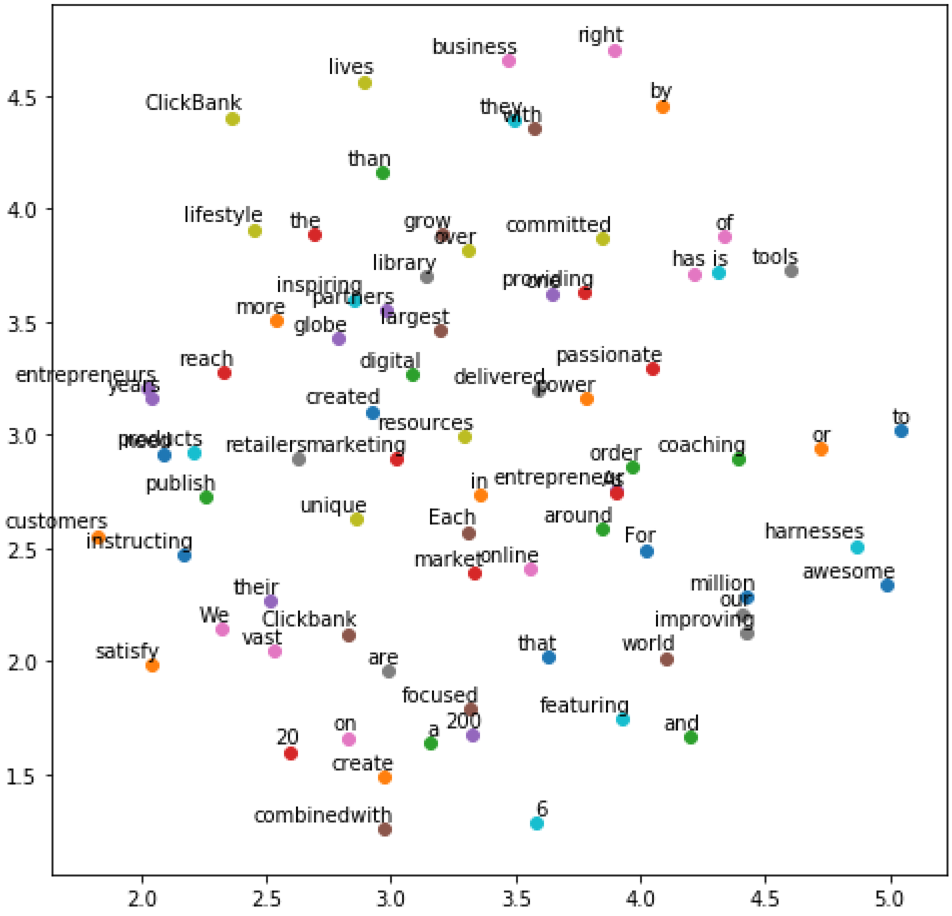
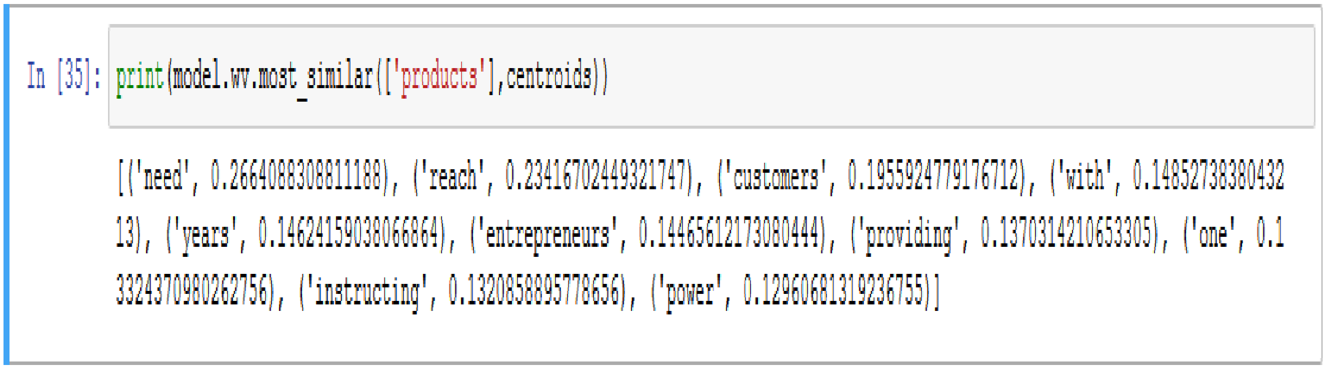


Figure 4.5: Plotted Clusters

- Then we found out the words which are nearest to their respective cluster.



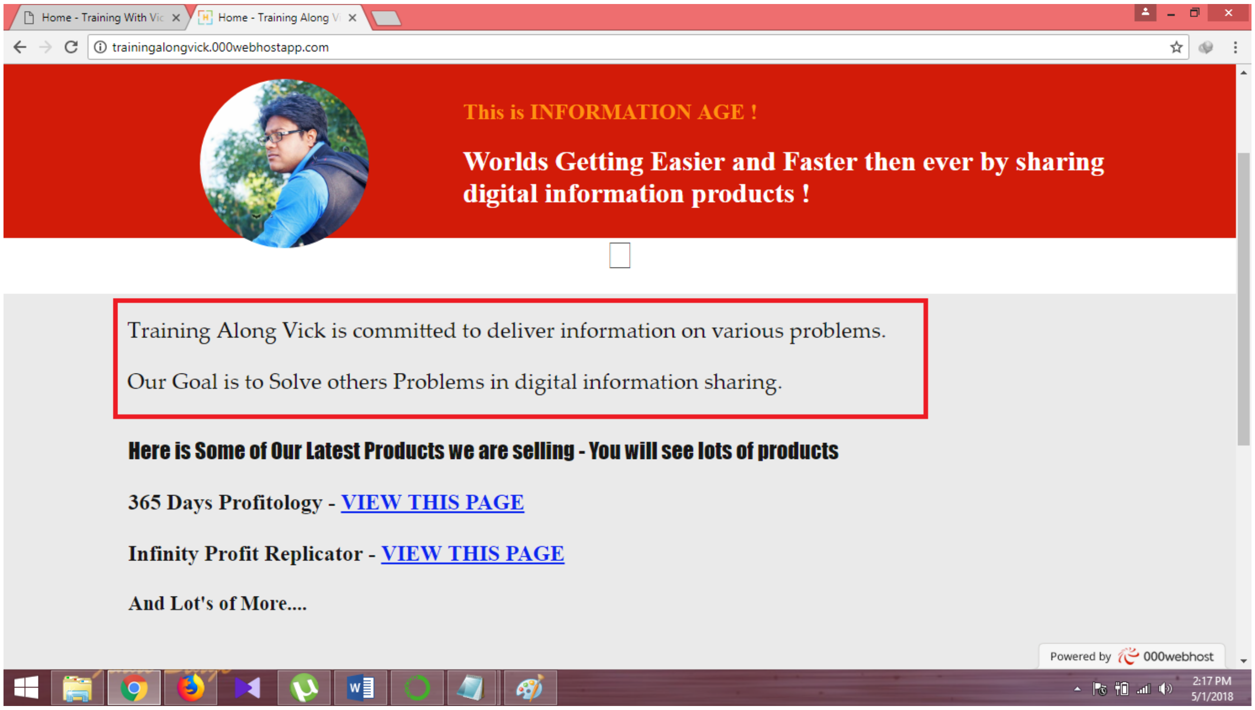
```
In [35]: print(model.wv.most_similar(['products'],centroids))

         [('need', 0.2664088308811188), ('reach', 0.23416702449321747), ('customers', 0.1955924779176712), ('with', 0.14852738380432
         13), ('years', 0.14624159038066864), ('entrepreneurs', 0.14465612173080444), ('providing', 0.1370314210653305), ('one', 0.1
         3324370980262756), ('instructing', 0.1320858895778656), ('power', 0.12960681319236755)]
```

Figure 4.6: Desired Output

4.3 Experimental Website

We experimented on two similar kind of website. One website is comparatively better than the other one.

The weaker website “trainingalongvick.000webhostapp.com” does not contain any Keyword in the title tag, body part and other sub section of the website. It does not contain any meta description either. Image optimization is also absent from this website.



This is INFORMATION AGE !

Worlds Getting Easier and Faster then ever by sharing digital information products !

Training Along Vick is committed to deliver information on various problems.

Our Goal is to Solve others Problems in digital information sharing.

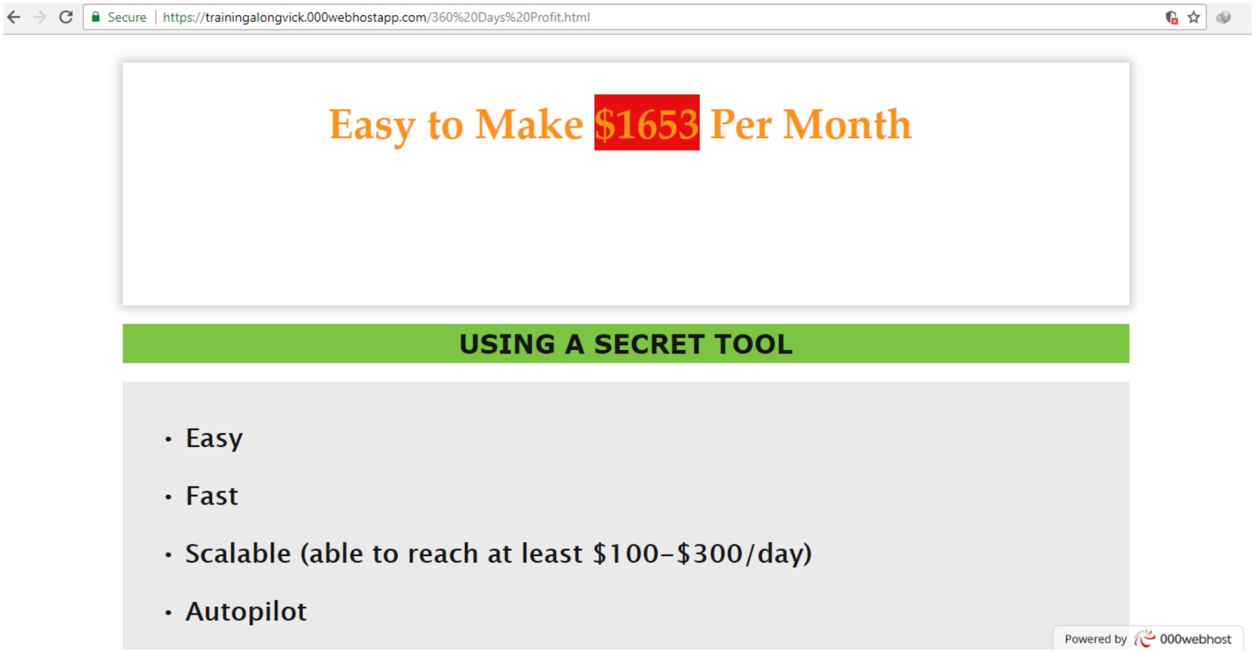
Here is Some of Our Latest Products we are selling - You will see lots of products

365 Days Profitology - VIEW THIS PAGE

Infinity Profit Replicator - VIEW THIS PAGE

And Lot's of More....

Figure 4.7: Body part and Header with less keywords and description

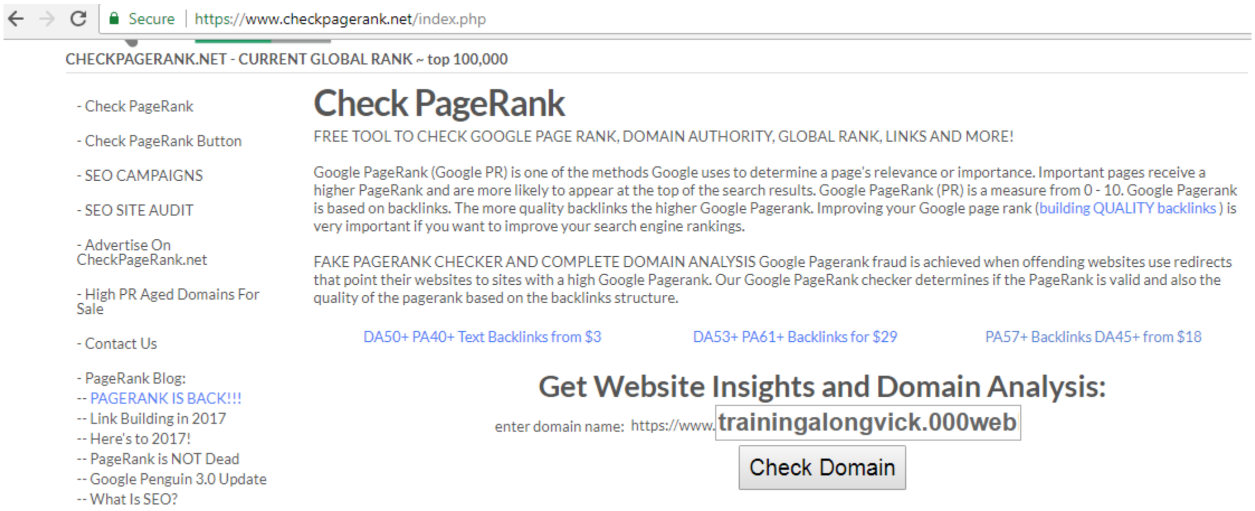

Secure
https://trainingalongvick.000webhostapp.com/360%20Days%20Profit.html
Powered by 000webhost

Easy to Make $1653 Per Month

USING A SECRET TOOL

- Easy
- Fast
- Scalable (able to reach at least $100-$300/day)
- Autopilot

Figure 4.8: Subsection with lesser details



Secure | https://www.checkpagerank.net/index.php

CHECKPAGERANK.NET - CURRENT GLOBAL RANK ~ top 100,000

- Check PageRank
- Check PageRank Button
- SEO CAMPAIGNS
- SEO SITE AUDIT
- Advertise On CheckPageRank.net
- High PR Aged Domains For Sale
- Contact Us
- PageRank Blog:
-- PAGERANK IS BACK!!!
-- Link Building in 2017
-- Here's to 2017!
-- PageRank is NOT Dead
-- Google Penguin 3.0 Update
-- What Is SEO?

Check PageRank

FREE TOOL TO CHECK GOOGLE PAGE RANK, DOMAIN AUTHORITY, GLOBAL RANK, LINKS AND MORE!

Google PageRank (Google PR) is one of the methods Google uses to determine a page's relevance or importance. Important pages receive a higher PageRank and are more likely to appear at the top of the search results. Google PageRank (PR) is a measure from 0 - 10. Google Pagerank is based on backlinks. The more quality backlinks the higher Google Pagerank. Improving your Google page rank (building QUALITY backlinks) is very important if you want to improve your search engine rankings.

FAKE PAGERANK CHECKER AND COMPLETE DOMAIN ANALYSIS Google Pagerank fraud is achieved when offending websites use redirects that point their websites to sites with a high Google Pagerank. Our Google PageRank checker determines if the PageRank is valid and also the quality of the pagerank based on the backlinks structure.

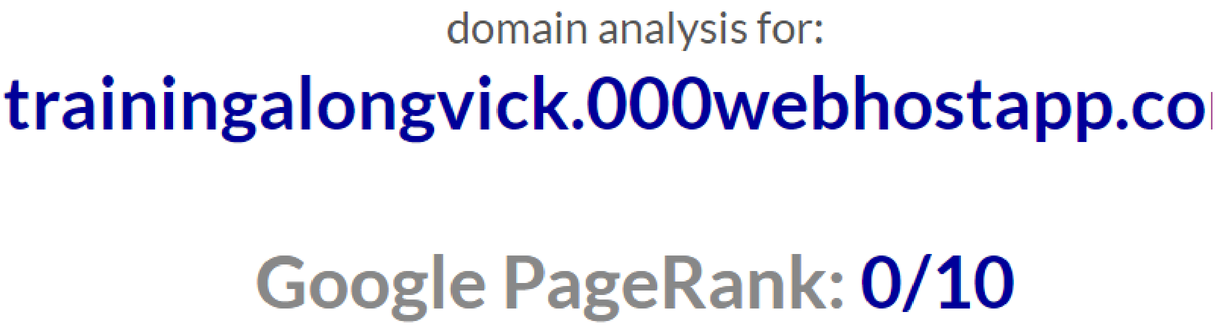
DA50+ PA40+ Text Backlinks from $3 DA53+ PA61+ Backlinks for $29 PA57+ Backlinks DA45+ from $18

Get Website Insights and Domain Analysis:

enter domain name: https://www.trainingalongvick.000web

Check Domain

Figure 4.9: Checking the page rank of Weaker website

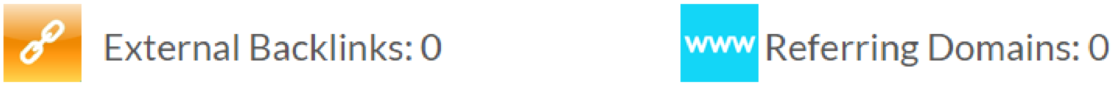


domain analysis for:

trainingalongvick.000webhostapp.co

Google PageRank: 0/10

Figure 4.10: PageRank of the weaker website

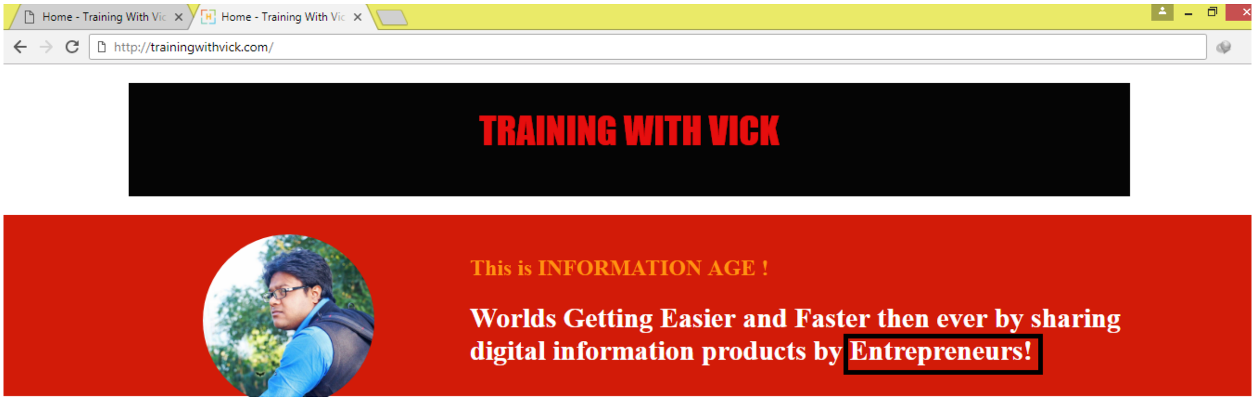


External Backlinks: 0

Referring Domains: 0

Figure 4.11: No. of external links and referring domains of weaker website

In the case of better website, we included our target keywords in the meta descriptions. The body of the website and the subsection were properly injected with keywords. All the subsection title header was wrapped inside header tag

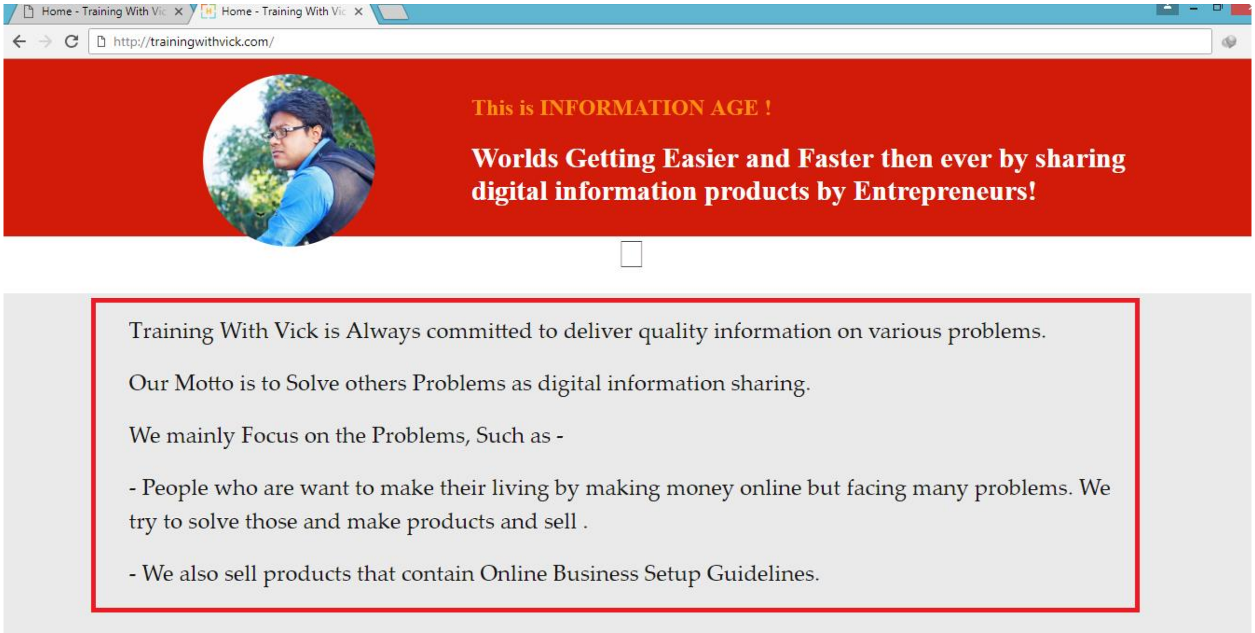


TRAINING WITH VICK

This is INFORMATION AGE !

Worlds Getting Easier and Faster then ever by sharing digital information products by Entrepreneurs!

Figure 4.12: Header filled with keyword



This is INFORMATION AGE !

Worlds Getting Easier and Faster then ever by sharing digital information products by Entrepreneurs!

Training With Vick is Always committed to deliver quality information on various problems.

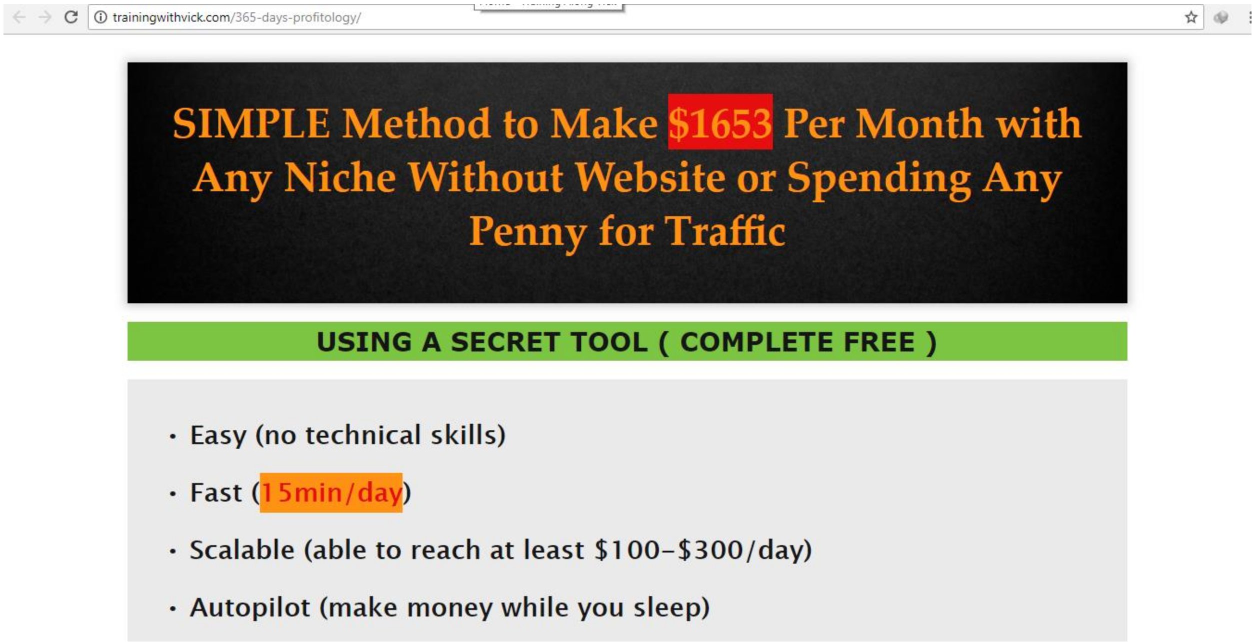
Our Motto is to Solve others Problems as digital information sharing.

We mainly Focus on the Problems, Such as -

- People who are want to make their living by making money online but facing many problems. We try to solve those and make products and sell .

- We also sell products that contain Online Business Setup Guidelines.

Figure 4.13: Enriched body part with more keywords

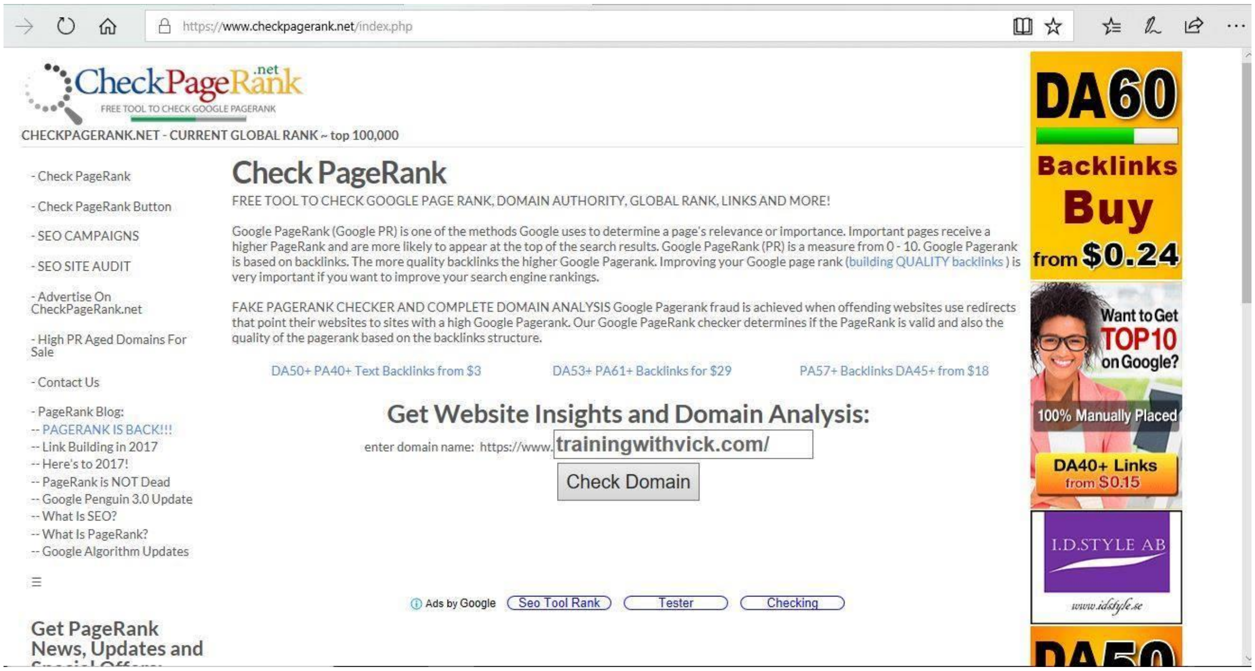


SIMPLE Method to Make $1653 Per Month with Any Niche Without Website or Spending Any Penny for Traffic

USING A SECRET TOOL (COMPLETE FREE)

- Easy (no technical skills)
- Fast (15min/day)
- Scalable (able to reach at least $100-$300/day)
- Autopilot (make money while you sleep)

Figure 4.14: Subsection and header with more keywords and more detailed


https://www.checkpagerank.net/index.php
CheckPageRank.net
FREE TOOL TO CHECK GOOGLE PAGERANK

CHECKPAGERANK.NET - CURRENT GLOBAL RANK ~ top 100,000

Check PageRank

FREE TOOL TO CHECK GOOGLE PAGE RANK, DOMAIN AUTHORITY, GLOBAL RANK, LINKS AND MORE!

Google PageRank (Google PR) is one of the methods Google uses to determine a page's relevance or importance. Important pages receive a higher PageRank and are more likely to appear at the top of the search results. Google PageRank (PR) is a measure from 0 - 10. Google Pagerank is based on backlinks. The more quality backlinks the higher Google Pagerank. Improving your Google page rank (building QUALITY backlinks) is very important if you want to improve your search engine rankings.

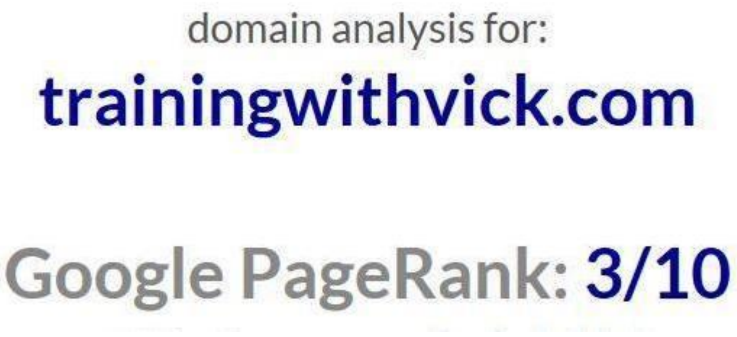
FAKE PAGERANK CHECKER AND COMPLETE DOMAIN ANALYSIS Google Pagerank fraud is achieved when offending websites use redirects that point their websites to sites with a high Google Pagerank. Our Google PageRank checker determines if the PageRank is valid and also the quality of the pagerank based on the backlinks structure.

DA50+ PA40+ Text Backlinks from $3
DA53+ PA61+ Backlinks for $29
PA57+ Backlinks DA45+ from $18

Get Website Insights and Domain Analysis:

enter domain name: https://www. trainingwithvick.com/

Figure 4.15: Checking the page rank of Good website

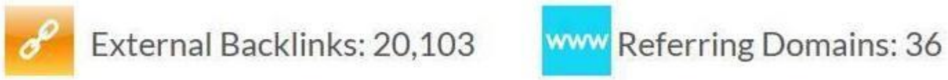


domain analysis for:

trainingwithvick.com

Google PageRank: 3/10

Figure 4.16: PageRank of the Good website



External Backlinks: 20,103

Referring Domains: 36

Figure 4.17: No. of external links and referring domains of good website

Chapter 5

Conclusion

In this thesis paper, we studied different PageRank Algorithms. Then we found out how to find PageRank of a web page using Sample Web Graph. Then we found out the problem of Google Matrix and we have solved the problem. Then we formed a general algorithm for PageRank.

We have studied about Search Engine Optimization (SEO) techniques, both off-page and on-page technique. Then we created two experimental websites and applied some of the techniques in one of them which we called the better website. Like we have used Keywords in title tag, meta description etc. and other techniques also. After applying those techniques we got the desired output which was to get PageRank of our better website.

Finally, we have implemented Word2Vec. We have used this to make the keyword research more efficient. We found out better keywords with this and applied those keywords in our better website.

We tried our best to apply all the techniques to get a better result and in future we will try to find out if there are more efficient ways to further improve the PageRank of our better website.

References

[1] Sergey Brin and Lawrence Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Seventh International World-Wide Web Conference (WWW 1998), Brisbane, Australia, April 14-18, 1998.

[2] Sepandar Kamvar, “Adaptive Methods for the Computation of PageRank”, Conference on the Numerical Solution of Markov Chains, 2003.

[3] A.K Sharma, “A Comparative Analysis of Web Page Ranking Algorithms”, (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 08, 2010, 2670-2676

[4] Wenpu Xing, “Weighted PageRank Algorithm”, CNSR '04 Proceedings of the Second Annual Conference on Communication Networks and Services Research Pages 305-314, May 19 - 21, 2004.

[5] https://en.wikipedia.org/wiki/URL, last updated: 1st March, 2018

[6] https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors, last updated: 1st March, 2018

[7] https://en.wikipedia.org/wiki/Power_iteration, last updated: 1st March, 2018

[8] https://en.wikipedia.org/wiki/Sparse_matrix, last updated: 6th March, 2018

[9] https://www.reliablesoft.net/what-is-off-page-seo, last updated: 6th March, 2018

[10] https://backlinko.com/google-ranking-factors, last updated: 6th March, 2018

[11] https://www.youtube.com/watch?v=E9aoTVmQvok&index=10&list=PLLssT5z_DsK9JDLcT8T62VtzwyW9LNepV, last updated: 26/04/2016

[12] Professor Brian A. Davey, “A guide for teachers – Years 11 and 12, Google PageRank”, Australian Mathematical Science Institute, 2013.

[13] https://en.wikipedia.org/wiki/K-means_clustering, last updated: 1st March, 2018

[14] https://en.wikipedia.org/wiki/Scikit-learn, last updated: 1st March, 2018

[15] http://scikit-learn.org/stable/modules/clustering.html#clustering,
Last updated: 1st March, 2018

Appendix A

Searching Keywords

This Appendix contains the code of how to search for the best keywords to use in the website. We used K-means Clustering for this and Python as our Programming Language.

1.

```
import gensim

from gensim.models import Word2Vec

from nltk.cluster import KMeansClusterer

import nltk

from sklearn import cluster

from sklearn import metrics

from sklearn.cluster import KMeans

%matplotlib inline

from sklearn.manifold import TSNE

import numpy as np

import matplotlib.pyplot as plt
```

2.

```
sentences = [['For', 'more', 'than', '20', 'years', 'Clickbank', 'has', 'delivered'],

             ['lifestyle', 'products', 'to', 'customers', 'around', 'the', 'globe'],

             ['Each', 'of', 'our', 'unique', 'products', 'is', 'created', 'by', 'a',
    'passionate', 'entrepreneur', 'focused', 'on', 'improving', 'the', 'lives', 'of', 'our',
    'customers', 'by', 'inspiring', 'instructing', 'or', 'coaching'],

             ['As', 'one', 'of', 'the', 'largest', 'online', 'retailers', 'ClickBank',
    'harnesses', 'the', 'awesome', 'power', 'of', 'our', 'digital', 'marketing',
    'partners', 'combined' 'with', 'a', 'vast', 'library', 'of', 'over', '6', 'million',
    'unique', 'products', 'in', 'order', 'to', 'reach', '200', 'million', 'customers',
    'around', 'the', 'world'],
             ['We', 'are', 'committed', 'to', 'featuring', 'products', 'that', 'satisfy',
    'our', 'customers', 'and', 'providing', 'entrepreneurs', 'with', 'the', 'right',
    'tools', 'and', 'resources', 'they', 'need', 'to', 'create', 'publish', 'and', 'market',
    'their', 'products', 'in', 'order', 'to', 'grow', 'their', 'business']]
```

3.

```
model = gensim.models.Word2Vec(sentences, min_count=1)
```

4.

```
print (model.wv.vocab)
```

5.

```
for word in model.wv.vocab:
   print(word,model[word])
```

6.

```
X = model[model.wv.vocab]

from nltk.cluster import KMeansClusterer
import nltk

NUM_CLUSTERS = 10

Kclusterer=KMeansClusterer(NUM_CLUSTERS,
distance=nltk.cluster.util.cosine_distance, repeats=25)

assigned_clusters = kclusterer.cluster(X, assign_clusters=True)

print(assigned_clusters)
```

7.

```
words = list(model.wv.vocab)

for i, word in enumerate(words):
   print (word + ":" + str(assigned_clusters[i]))
```

8.

```
kmeans = cluster.KMeans(n_clusters=NUM_CLUSTERS)
kmeans.fit(X)

labels = kmeans.labels_

centroids = kmeans.cluster_centers_

print ("Cluster id labels for inputted data")

print (labels)

print ("Centroids data")

print (centroids)

print ("Score (Opposite of the value of X on the K-means objective which is
Sum of distances of samples to their closest cluster center):")

print (kmeans.score(X))

silhouette_score = metrics.silhouette_score(X, labels, metric='euclidean')

print ("Silhouette_score: ")

print (silhouette_score)
```

9.

```
# define the function to compute the dimensionality reduction
# and then produce the biplot

def tsne_plot(model):
   "Creates a TSNE model and plots it"
   labels = []
   tokens = []

   for word in model.wv.vocab:
      tokens.append(model[word])
      labels.append(word)

   tsne_model = TSNE(perplexity=40, n_components=2, init='pca',
   n_iter=2500)
   new_values = tsne_model.fit_transform(tokens)
```

```
    x = []
    y = []

    for value in new_values:

        x.append(value[0])
        y.append(value[1])

    plt.figure(figsize=(8, 8))

    for i in range(len(x)):
        plt.scatter(x[i],y[i])
        plt.annotate(labels[i],
                     xy=(x[i], y[i]),
                     xytext=(5, 2),
                     textcoords='offset points',
                     ha='right',
                     va='bottom')

    plt.show()

# call the function on our dataset
tsne_plot(model)
```

10. print(model.wv.most_similar(['products'],centroids))