Links Analysis of Wikipedia Documents using MapReduce

Vasa Hardik   Vasudevan Anirudh   Palanisamy Balaji

School of Information Sciences, University of Pittsburgh
{hnv3, anv52, bpalan}@pitt.edu

***Abstract*—Wikipedia, a collaborative and user driven encyclopedia is considered to be the largest content thesaurus on the web, expanding into a massive database housing a huge amount of information. In this paper, we present the design and implementation of a MapReduce-based Wikipedia link analysis system that provides a hierarchical examination of document connectivity in Wikipedia and captures the semantic relationships between the articles. Our system consists of a Wikipedia crawler, a MapReduce-based distributed parser and the link analysis techniques. The results produced by this study are then modelled to the web Key Performance Indicators (KPIs) for link-structure interpretation. We find that Wikipedia has a remarkable capability as a corpus for content correlation with respect to connectivity among articles. Link Analysis and Semantic Structuration of Wikipedia not only provides an ergonomic report of tire-based link hierarchy of Wikipedia articles but also reflects the general cognition on semantic relationship between them. The results of our analysis are aimed at providing valuable insights on evaluating the accuracy and the content scalability of Wikipedia through its link schematics.**

I. Introduction

Wikipedia, a collaborative and user driven encyclopedia is considered to be the largest content thesaurus on the web, expanding into a massive database housing a huge amount of information. English Wikipedia contains more than 4.4 million articles which includes content from wide range of concepts on various fields including Arts, Geography, History, Science, Sports and Games. This massive graph model of Wikipedia contains billions of link edges that connect one article to the other. Link analysis of such huge web corpus helps in determining how well the articles are connected to each other. Some of the basic postulates that differentiates Wikipedia from other conventional web documents include [1]:

- First, Wikipedia link structure is similar to the Web, but it is a more densely connected graph.
- Second, unlike the web, Wikipedia's outbound links have high semantic similarity to the inbound links and they carry equal importance.
- Third, the structure of Wikipedia suggests that it is designed/has evolved in such a way in order to provide higher emphasis to internal link traversals, confined to its own link hierarchy.

Hyperlinks in web documents indicate content relativity, relatedness and connectivity among the linked articles. In our work, we use semantic relatedness as a numerical strength of relationship between two web documents [2]. For example if most of the top links from an Wikipedia article 'Epidemic' points to the article 'Ebola', it shows that these two subjects are highly related. Similarly if the article 'Music' is strongly linked to the article 'Rap', it denotes the popularity of rap music. When handling large scale web documents, distributed data storage has become an efficient solution for applications that require high efficiency and scalability. This type of storage allows parallel data processing across multiple distributed computing nodes. To achieve large scale processing of Wikipedia dumps in our analysis, we adopt a cloud-based solution using MapReduce. In phase one of our system, a Python based web crawler, *Cleoria* is designed to crawl and store Wikipedia articles as raw dumps. In the second phase, a MapReduce-based dynamic parser is used to parse the web documents from the collected dumps. From the parsed documents, statistical analysis of the connectivity and relationship among the articles in the document is performed. As a corpus for knowledge extraction, Wikipedia's impressive characteristics include not only its scale but also to its dense link structure. These characteristics are valuable to extract accurate knowledge trends from Wikipedia.

This is characterized by a web KPI called Semantic Structuration that studies the Wikipedia Ontology [3] of link structure and denotes the required manipulation that can be done in a given article to increase its efficiency. It also brings into consideration the effect on the overall Wikipedia model by bringing about these changes. Wikipedia's links are promising resources for its ontology structuration [4]. In this paper, we propose a semi-automated design model for link analysis by mining hyperlinks from articles that help us in content relation interpretation as links between web pages can be viewed as indicative of the quality and importance of the pages they point to. In our study, we also shed light to the unique characteristics of Wikipedia web link structures with respect to their value as relevant indicators of a page for a given topic of request. This analysis has taken into consideration Gigabytes of data from the Wikipedia dumps, covering tens of thousands of articles and millions of lines of raw XML text.

The rest of the paper is organized as follows. In Section 2, we present the design overview and architecture of the data distribution. We also discuss the implementation of the *Cleoria* web based crawler and techniques to parse the crawled data based on MapReduce. We discuss the results of various analysis in Section 3. We discuss related work in Section 4 and we conclude in Section 5.

II. Design Overview

The schematics of our design include the development of a Wikipedia based web crawler and a MapReduce parser. We use Hadoop Distributed File System (HDFS) as the storage platform used for implementing the proposed system as it provides the desired scalability for processing [5]. The design of our system requires two separate components: the first component is responsible for collecting data using the web crawler and the second is responsible for parsing and processing the obtained data.

A. Data Acquisition

With the growing size of the Wikipedia database, the connectivity among its entities is important, which in terms of web ontology is provided by the page hyperlinks [6]. These hyperlinks are sub divided primarily into internal hyperlinks (hyperlinks that link to another Wikipedia article) and external hyperlinks (hyperlinks that link to web pages outside of the Wikipedia domain). Internal and External links are important performance parameters to determine robustness of Wikipedia [7].

To obtain a data set consisting of the English Wikipedia articles, we have designed and developed a high speed web page crawler named *Cleoria*. This is a Wikipedia bot (web spider) that systematically scans the articles for the purpose of dump collection. Along with accessing the web page, *Cleoria* also downloads the content of the page [8] in the XML format for performing parsing operation. The system is optimized to achieve low latency where up to 10 pages per second can be crawled and downloaded on a single virtual machine (VM) and this linearly decreases with increase in the number of virtual machines used.

Algorithm 1 Python Based Wikipedia Crawler Algorithm

```
function $web\_crawl$
    $to\_crawl$ initiate array with $starting\_page$
    crawled <- initiate crawled as an empty array
    i=0
    while True do
        urll <– pop the first url
        urll,flag <- $url\_parse(urll)$
        flag2 <- check for non web documents
        if flag <- 1 or flag2 <- 1 then
            pass
        else
            if urll is in crawled then
                pass
            else
                $raw\_html$ <- $web\_page$
                $see\_also$,flag2 <- extract $see\_also$ section
                $raw\_intro$ <- extract intro
                $to\_crawl$ <- $to\_crawl$ + $get\_all\_links$
                crawled.append(urll)
                $pure\_intro$ <- extract intro
                database [title] <- $pure\_intro$
                file <- write data to the file
                remove duplicated from $to\_crawl$ list
            end if
        end if
    end while
    return ””
end function
```

The web crawling procedure of the *Cleoria* web-bot starts with a set of URLs to visit. This set of URL is referred to as the seed page. The bot initializes by downloading an entire web page as a raw XML document. The implementation of the bot is such that it extracts all the hyperlinks from the downloaded XML document. Once extracted, an array of hyperlinks is populated and stored locally for further processing. In the next step, each element of the array is a hyperlink which is given to the URL normalizer to convert relative URLs into their corresponding absolute form. Additionally the normalizer also verifies if each hyperlink's network location (Netloc) is same as the 'Netloc' of the seed page. Network location is the domain name in the web URL but does not include the scheme (for example, 'http') and the path (directory structure). For example if the hyperlink is not part of the Wikipedia domain then the normalizer simply discards it.

However before storing the hyperlink into the file system, it checks for its possible duplication. On the occurrence of duplication, the specific hyperlink is discarded from the populated list. This is an efficient technique which reduces the overhead time of crawling the same URL repeatedly. Upon storing the URLs, the above processes concurrently repeat until the entire web corpus is crawled. *Cleoria* uses the breadth-first mode of crawling its web frontier which reduces the processing overhead and also increases the speed of crawling. Once the Wikipedia dump is created by *Cleoria*, further processing and analysis of the obtained data is carried out as we describe in the next subsection.

B. Data Processing

This component of our system focuses on extracting the data from the web crawler and performing parsing process. The concept of web scrapping is highlighted in this section which is modified with respect to our cloud based model. By definition, Web scraper is a computer software technique of extracting information from websites [9].

In our system, we use web scrapping with an integration of MapReduce which has helped us model the parser, which essentially extracts the content from the crawler phase and separately runs the map and reduce operations for the segregation of various types of hyperlinks. The data obtained after the crawler is unstructured, and this MapReduce parser transforms it into a structured format and stores the output back to HDFS. Wikipedia has an existence of a wide variety of hyperlinks which helps it connect with different domains of the web. These categories of hyperlinks can be differentiated in accordance to their link structure and syntax. In context to our design the MapReduce parser can differentiate the following categories of hyperlinks enlisted in Table 1.

While loading the data into the HDFS, the collective Wikipedia dump is divided into smaller chunks of 64MB splits where each split contains numerous Wikipedia articles on which the parsing operation is to be carried out. Each web page is identified as a separate entity with the help of its unique page ID(as that of Wikipedia Articles). The initial step of web parsing (the first step of the map phase) includes scanning the webpage and extracting all the hyperlinks from it. On each split, a map process is executed.

Algorithm 2 MapReduce Based Wikipedia Parser Algorithm

```
function web_parse
    for line in sysṡtdin do
        if  wgArticleId in line then
            key <- 'Articles'
            value <- 1
            print(key, value)
        else
             links <- get_all_links from line
            for j in links do
                if ' thenhref' in line:
                    s <- line.find('href')
                    if  '.jpg' in line or '.png' in line then
                        key <- 'Image Links'
                        value <- 1
                        print(key, value)
                    else if 'en.wikipedia.org' in line then
                        key <- 'Internal but Irrelevant'
                        value <- 1
                        print(key, value)
                    else if '.wikipedia.org' in line then
                        key <- 'Non-English Wiki Links'
                        value <- 1
                        print(key, value)
                    else if  'wikimedia.org' in line then
                        key <- 'Organizational Link'
                        value <- 1
                        print(key, value)
                    else if  '/wiki/' in line[s+6:s+15] then
                        key <- 'Internal Link'
                        value <- 1
                        print(key, value)
                    else
                        key <- 'External Link'
                        value <- 1
                        print(key, value)
                    end if
                else
                    pass
                end if
            end for
        end if
    end for
end function
```

The map function is used to find out the occurrences of string 'href' (hyperlink reference) in a given split. It then searches for the specific string values to differentiate between different categories of hyperlinks as mentioned in Table 1. The map job will have the link category as the key and its value as '1'. With the completion of the map phase, all the segregated hyperlinks are given to the reduce phase. The reducer combines all the hyperlinks given by the mappers and sums up the URL count per category. This concludes the web parsing process which gives the total URL count for each category as the output.

Name	**Notation**	**Properties**	**Structure**
Internal Relevant	IR	Links articles within Wikipedia	/wiki/
Internal Irrelevant	INR	Links articles within Wikipedia but not articles	/w/ or en.wikipedia.org
External	EXT	Links articles with external domains	//creativecommons.org
Organization	ORG	Links articles to the organizational entity (Wikimedia)	/wikimedia.org/ or /wikimediafoundation.org/
Non-English articles	NOE	Non-English Wikipedia links	.wikipedia.org
Images	IMG	Links that represent images	.png, .jpeg, .jpg, .tiff, .svg, .xcf

TABLE I: Link Categories

III. Analysis

In this section we present our experimental analysis on a test bed to illustrate the characteristics of article hyperlinks, their accuracy, linkage and relevance. For our experiments, the evaluation was carried on a Hadoop cluster of two virtual machines. The virtual machines were configured with Ubuntu 14.04.1 LTS [10]. The frameworks that were installed include Java Virtual Machine (JVM), Python 2.7 [11], Apache Hadoop 2.6.0 and MapReduce. The dependencies for our design included various sizes of the Wikipedia dumps consisting of raw XML documents.

A. Code Accuracy

Web parsing is a process that involves semantically defragmenting a web page and extracting relevant information from it. Various iterative techniques were implemented to achieve high accuracy in hyperlink extraction. Computation of parser includes iterative techniques like recursive scanning, parsing stripped lines and parse clearance.

From the graph shown in Figure 1 we can observe the accuracy of the MapReduce application with respect to the average number of links computed per article. This analysis shows that on an average there are about 657 links present in a page. Recursive scanning enables to scan a given web page more than once and in each scan operation it finds the section of the web page from which the hyperlinks are to be extracted. Any given article contains stripped lines that are programmatically beyond one line break. This extended line contains more than one hyperlink and it is important to extract every hyperlink. Parse clearance helps us discard the unwanted scripts in a given web document, this includes inbuilt style sheet and/or Java scripts. The introduction of these performance enhancement techniques are important from the prospective of our design model. With the inception of these techniques in an iterative manner the average code accuracy reaches to its maximum level. As evident from Figure 1 the maximum achievable code accuracy is about 99.8%.

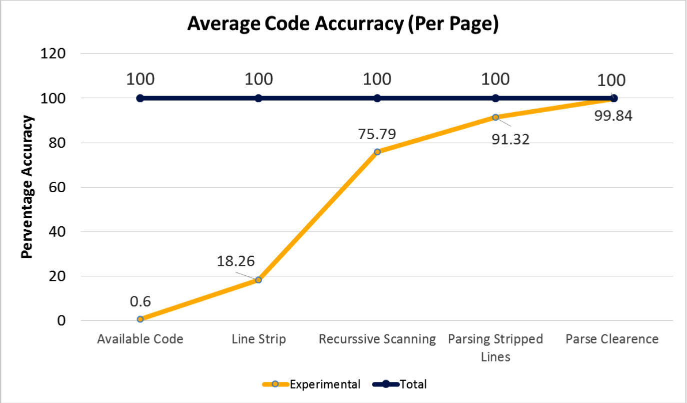

Fig. 1: Average Accuracy

B. Links Per Article

A large fraction of the Wikipedia dump was taken into consideration to test the average number of links per article. This database consisted of Wikipedia dump of 418MB and comprising of more than 3.2 Million code lines (approximately 5000 Wikipedia articles). The output from the MapReduce parser gives the average number of hyperlinks of various link categories. From the graph in Figure 2, we have calculated the average number of links in different categories as mentioned in Table 1. The number of image hyperlinks seen in the figure proves that Wikipedias web frontier is not only content heavy but also relatively image heavy. Furthermore it is also evident that the average number of internal links is almost twice as much as the external links. This helps us in evaluating the interconnectivity of articles within the Wikipedia domain.

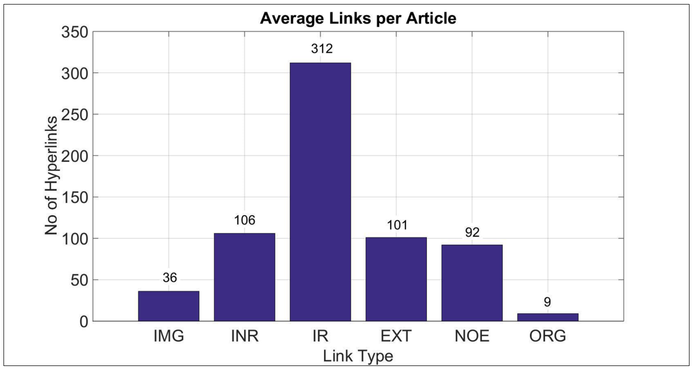

Fig. 2: Average number of links per article

Another important observation is that apart from the English language, Wikipedia also supports the information of the same page in almost 92 different languages. And there is a strong linkage of an article to its corresponding article in other languages. The hierarchical interconnectivity is also seen as each article is directly connected to the governing bodies of Wikipedia 'The Wikimedia' and 'The Wikimedia Foundation'. This is constantly observed in all the articles which denotes that Wikipedia uses a standard connectivity template. The presence of hyperlinks that are not linked to any other article is also observed. These hyperlinks are denoted as external hyperlinks and provide additional interoperability between the backend database and the articles. External connectivity of each article with a domain apart from Wikipedia is an important parameter. This is illustrated by the statistical result from the figure where each article is at least connected to more than 100 external references which in turn increases its diversity. We can see that Wikipedia is a well-connected network with an almost 2:1 ratio in context to internal and external domains. On persistent computation of the data set, we have observed an average figure in the same range which emphasizes the integrity and accuracy of the framework as well as the data set [12].

C. Link Comparison

A different prospect of analyzing hyperlinks is by categorizing the articles. Wikipedia consist of more than 40 categories of articles. For this study six random domains - sports, film, biology, technology, health and education were chosen. From these domains a set of popular articles were choosen (again at random) and average internal and external hyperlinks were computed. The result suggest the relation between the number of Internal links to that of the external links. The strength of a particular category in terms of its semantic relatedness exist if the total number of internal links exceeds the total number of external links. The internal and external hyperlink ratio varies according to the domain type. From the graph in Figure 3, it can be observed that categories like sports, technology, health and education the internal hyperlinks are more in number than the external ones, this indicates that these categories are more internally dependent. Whereas categories like film and biology are more externally influenced. The above result demonstrates the high dynamic nature of Wikipedia link structure.

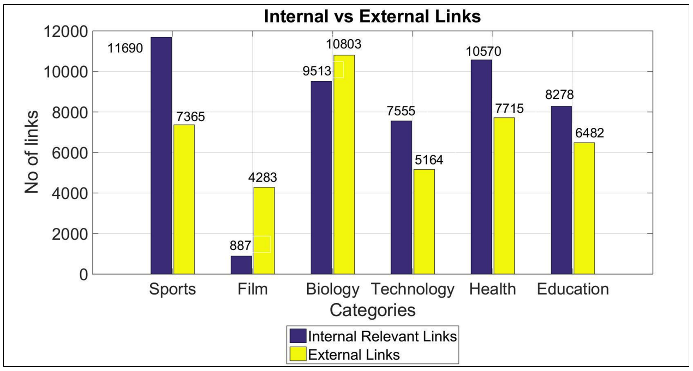

Fig. 3: Link Comparison

D. Link-ability Factor

The link-ability factor is the numerical representation of number plausible hyperlinks in a given Wikipedia article.

The link-ability factor helps us determine the diversity and expandability of Wikipedia. This can be explained commonly in terms of link efficiency of the article [13]. Higher the link-ability factor of an article, greater is its content connectivity. Our design model defines link-ability factor as a ratio of total internal relevant links to the total unique words in an article. Mathematically this can be computed as follows:

$$Link-abilityFactor(LF)=\frac{\sum l-\sum I_l}{Unique(\sum(W-S_W)))}$$

where, l is total links in an article, I_l is external links in an article, W is total words and S_W is stop words. For example, when considering the Wikipedia article titled 'Resistor', the achieved Link-ability Factor is 2.805%, whereas that of the article 'Earth' has Link-ability Factor of 20.0145%. This implies the article 'Earth' has significantly higher amount of links per words and hence determines greater connectivity. Overall, Link-ability factor of Wikipedia articles varies from 0.2% to as high as 25%. It denotes a numeric variable of total words linked in an article to the unique words in an article that could be linked. If all the words in an article that can be linked, are actually linked, then the Link-Ability factor converged to one (100%). To increase the Link-ability Factor, more number of words needs to be linked to the internal Wikipedia articles. This linking of words needs to be done in a systematic way considering the wiki markup heat map process. On an average, the Link-ability factor of the entire Wikipedia is about 11-12%. This shows there is massive scope of improvement in the LF [14]. Increase in LF significantly increases the robustness of Wikipedia.

E. 3D Wikipedia graph model

The visualization of domain can be expressed in a 3 dimensional space. This graph model can be considered with articles as nodes and their connectivity links as edges. This 3D model and be envisioned in the form of a sphere. The placement of the articles in this sphere can be determined in accordance to their relative link-ability factor [15]. The articles having higher link-ability factor will be positioned towards the core or centre of the sphere whereas the articles with least link-ability factor towards the periphery.

Ideally for Wikipedia to be 100% efficient the link-ability factor of every article should be one (on a scale of 0-1). In such a condition the 3D graph would converge to a 'point' instead of a sphere. Attaining such an efficient system, all the articles must be completely link-able. However, practically this is difficult to realize given the dynamic nature of the content flow in and out of Wikipedia. With the present model the overall system connectivity can be improvised by using two approaches. First by increasing the connectivity of the core and make it as dense and connected as possible. This would make the core heavy and semantically pull the articles nearer to the core. Second, by increasing the connectivity of the periphery articles so that the drift is directly towards the core. Impact of these approaches towards the overall system is different and the challenge is to select the method which will increase the connectivity efficiently.

According to the relatedness and connectivity metrics, one can postulate an efficient approach to increase the frontier of Wikipedia by increasing the link-ability factor of the nodes in the periphery [16]. This approach will provide a consistent and steady change, where the load on the core articles of nodes will not increase drastically. Also, articles can be made more inclined to the epicentre. This approach will additionally provide better assistance towards content distribution in the Wikipedia corpus which can help the organization place and link their articles effectively.

F. Article Relevancy

Articles relevancy is a web KPI which determines the congruency of any linked web document. With respect to Wikipedia, this is an essential parameter which affirms the semantic relatedness between articles [17]. Additionally effectiveness of an article is denoted by its linkage to all possible articles in the context and less to the article not in the given context. This can be manually concluded with the help of distance of one article with respect to the other in the 3D space. For a particular Wikipedia article, the relevancy metric is defined as the ratio of number of relevant article hyperlinks to number of irrelevant article hyperlinks. According to the analysis done for a specific Wikipedia article - 'University of Pittsburgh' on the links with higher occurrences, we have observed that the number of relevant links greatly exceed the

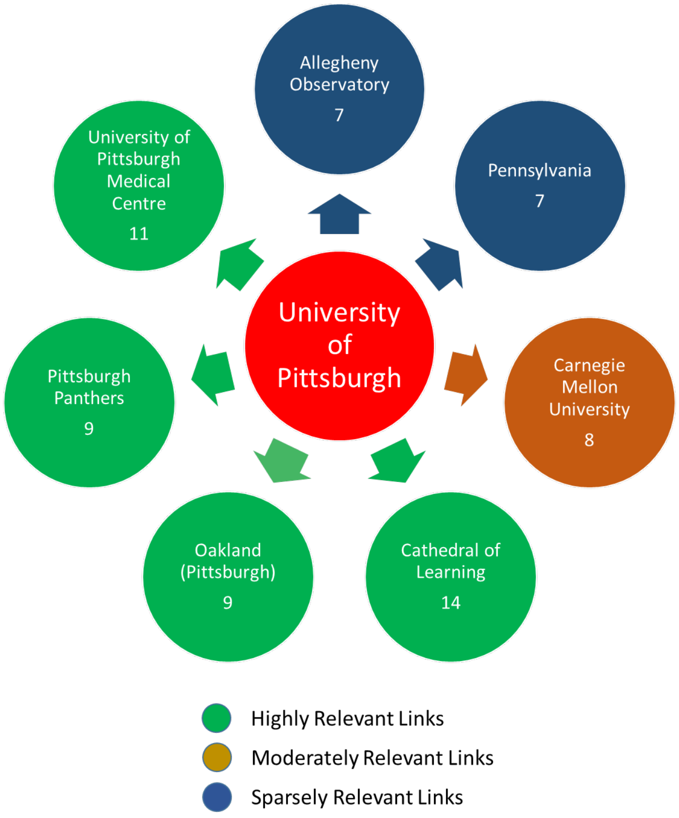

Fig. 4: Article Relevancy Metric

number of irrelevant or distant-relevant links as seen in Figure 4 [18]. We conclude that the scope of the article is immensely decisive on the type of links present in that article.

IV. Related Work

The foundation of our paper is based on the Link analysis of the English Wikipedia articles and to study their behavior with the use in the cloud based distributed systems. Similar link analysis structure was carried out in [1] where the comparison of Wikipedia and Web link structure was conducted. This work also illustrates the importance of in-links and out-links with reference to the article importance and how better can link analysis improve the performance of search engines.

The robustness and integrity of a search engine is a vital character which is explained in [13] where the correctness of Wikipedia is evaluated using link structure. A similar approach is adopted in our design where we also evaluate the link relevancy using the link structure of the articles. In the work described in [15], the differentiation on the basis of graph theory is compared using semantic relatedness of articles. However to modify this approach, we have used the MapReduce platform to substantiate the relatedness metric and determine the connectivity graph of Wikipedia articles with respect to their link-ability factor.

The work of Stephen Dolan has also contributed towards the implementation of our design in which he focused on the strength of connectivity among articles [19]. His research also is involved around finding the hop factor which denotes the number of clicks that is required to reach from one article to another within the Wikipedia domain. His work intuitively help us recognize the web KPIs and device the different design techniques discussed previously to formulate and realize our tests.

V. Conclusion

In this paper we propose a new method to semantically study and analyze the Wikipedia Links. For the same, we use a cloud platform to device a technique to compute the diversity and link-ability of Wikipedia articles. Wikipedia being one of the largest content data sets, it becomes an inquisitive platform to analyze and determine how well connected its articles are. Our design model also computes concept-content relations and concept-hyperlink relations to discover the semantic relationship between concepts within Wikipedia. It is also evident that the link structure ontology of Wikipedia is fundamentally different from that of the general web.

The evaluation results on concept and interpretation show that our method substantially outperforms approaches of the past. The results of our evaluation not only display the effectiveness of the link-ability factor algorithm but also quantifies the semantic relatedness among various categories of articles. The aim of this work was to build an algorithm that can be of signicant importance to increase the link efficiency of the Wikipedia domain model. With that in mind, the output of our algorithm provides enough information so as to prioritize the content manipulation to increase the overall link efficiency. To substantiate this claim, we described how measures of semantic relevancy can be adapted to evaluate the relatedness among the content.

VI. Future Work

In future we intend to elaborate the scope of this project not only to the complete Wikipedia corpus but to the entire domain. At that scale, higher accuracies will be achieved to the previously mentioned web KPIs. We also plan to explore new metrics for better visualization of the article hierarchy in the given 3-Dimensional Wiki Space. In another direction of our future work, we intend to study the impact of the distributed crawler by verifying the independent behavior of each phase of the MapReduce based crawling system to reduce the redundancy overhead while crawling.

References

[1] Jaap Kamps, Marijn Koolen. Is Wikipedia Link Structure Different?. In Proceeding of Second ACM International Conference on Web Search and Data Mining, Spain, 2009. (pp 232-241)

[2] Raiza Tamae Sarkis Hanada, Marco Cristo. How Do Metrics of Link Analysis Correlate to Quality, Relevance and Popularity in Wikipedia? In Proceedings of the 19th Brazilian symposium on Multimedia and the web (pp 105-112).

[3] Fan Bu, Yu Hao and Xiaoyan Zhu. Semantic Relationship Discovery with Wikipedia Structure. In Proceeding IJCAI'11 Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three (pp 1770-1775).

[4] Qi Zhang, Ling Liu, Kisung Lee, Yang Zhou. Improving Hadoop Service Provisioning in A Geographically Distributed Cloud. In Proceedings of the 2014 IEEE International Conference on Cloud Computing (pp 432-439). Dec 2011.

[5] Michele Nemschoff. Big data: 5 major advantages of Hadoop. http://www.itproportal.com/2013/12/20/big-data-5-major-advantages-of-hadoop/. Dec 2013.

[6] Kotaro Nakayama, Takahiro Hara and Shojiro Nishio. Wikipedia Link Structure and Text Mining for Semantic Relation Extraction. Proceedings of the Workshop on Semantic Search (SemSearch 2008) at the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, (pp 59-73)

[7] Vilie Farah. How Internal Links Affect PageRank, LINK EXCHANGE, April, 2009

[8] Wikipedia Authors Web Crawler. https://en.wikipedia.org/wiki/$Web\_crawler$

[9] Wikipedia Authors Web Scrapping. http://en.wikipedia.org/wiki/$Web\_scraping$

[10] Blog Authored. What Is Linux: An Overview of the Linux Operating System, April 2009.

[11] Michele Nemschoff. Java vs Python - Which Programming Language Should Learn First.

[12] Adrian M. Kentsch, Walter A. Kosters, Peter van der Putten and Frank W. Takes. Exploratory Recommendation using Wikipedias Linking Structure. Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands.

[13] Benjamin Mark Pateman, Colin Johnson. Using the Wikipedia Link Structure to Correct the Wikipedia Link Structure. Proceedings of the 2nd Workshop on Collaboratively Constructed Semantic Resources, Coling 2010, (pp 1018).

[14] David Milne and Ian H. Witten. Learning to Link with Wikipedia. Proceedings of the sixteenth conference on information and knowledge management CIKM'09 (pp 509-518).

[15] Torsten Zesch and Iryna Gurevych. Ubiquitous Knowledge Processing Group Telecooperation Division. Analysis of the Wikipedia Category Graph for NLP Applications. Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing 2007 (pp 1-8).

[16] David Liben-Nowell and Jon Kleinberg. The Link-Prediction Problem for Social Networks. Proceedings of the twelfth international conference on Information and knowledge management CIKM '03(pp 556-559).

[17] David Milne, Ian H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA, 13 July, 2008. (pp. 25-30).

[18] F. Bellomi and R. Bonato. Network Analysis for Wikipedia. Proceedings of Wikimania 2005, The First International Wikimedia Conference.

[19] Stephen Dolan. Six degrees of Wikipedia. http://mu.netsoc.ie/wiki/.