Weighted Semantic PageRank Using RDF Metadata on Hadoop

Hee-Gook Jun[1], Woo-Hyun Lee[1], Dong-Hyuk Im[2], Sang-Goo Lee[1], and Hyoung-Joo Kim[1]
[1] School of Computer Science and Engineering, Seoul National University, Seoul, Korea
[2] Dept. of Computer and Information Engineering, Hoseo University, Chungnam, Korea

Abstract *- PageRank, a representative link-based algorithm, evaluates the importance of Web pages based on the number of in-links each has. However, this feature may cause a problem in that pages with many in-links can be highly ranked regardless of their importance to the given query. Many methods have attempted to solve this problem by evaluating the weight of the links to stratify their importance. However, these methods have a limitation in that the weight of the links cannot be evaluated by their meaning directly owing to the hyperlink-based Web structure. We therefore propose a new approach to utilize the meaning of links directly by changing from a hyperlink-based Web structure to a semantic-link-based Web structure. In addition, we implemented the ranking method using the MapReduce framework to improve performance of semantic Big Data processing. The results of our experiment show that our approach outperforms the existing PageRank algorithm.*

Keywords: Big Data, MapReduce, Semantic Web, PageRank, RDF

1 Introduction

As the World Wide Web produces a greater amount of information over time, such information needs to be processed more effectively and efficiently to provide more accurate information to users. This issue has become a key challenge for Web-based information retrieval [8, 12, 19]. Since the 1990s, various methods dealing with the explosion of information on the Web have been studied in the field of Web information retrieval, including indexing, clustering, user interface methods, and ranking.

A page-ranking algorithm is an essential Web information retrieval method as the volume of results matched with a given query during a retrieval step is hard to be managed by users. Such an algorithm answers a given user query with a page list ranked by importance. The early page-ranking algorithm was a term-based ranking algorithm, whose criterion for evaluating the importance of pages is how many matched terms [3, 4] are contained on the page. After 1998, alternative ranking algorithms based on a linked relationship of pages [5, 11] were provided, and proved that link-based ranking algorithms perform better than term-based algorithms.

PageRank [5] is a representative link-based ranking algorithm. The authors of this algorithm assume that important pages are referred to by many other pages. Through this method, each page distributes its rank score to other pages they link to. Therefore, the more in-links a page has, the more important the page is. PageRank, however, which does not consider the semantics of links when computing their importance, may highly rank pages that only contain meaningless in-links.

Many algorithms have been suggested to tackle the above problem [18, 20, 26], and some have considered evaluating the weight of the links to adjust the propagation of their rank scores. In this way, if a page has many in-links with a small weight value, the page will have a lower value of importance. However, for a hyperlink-based Web structure, there are two significant limitations in evaluating the weight of a link. First, the hyperlink does not explain why pages are linked to other pages. Therefore, existing algorithms evaluate the weight of the links indirectly, such as by counting the number of links or analyzing other features out of links. Second, a page used as a unit of ranking is actually an object containing information rather than information itself. In other words, highly ranked pages that have high importance values owing to the presence of many in-links do not always contain important information, and may even contain meaningless information.

Moreover, the other problem of ranking is that ranking algorithms require a large space to store Web link structure and ranking values for every page. It is not easy for a single machine to compute large-scale data and to produce ranking results in a reasonable time. Hence searching an appropriate Big Data processing method for ranking is essential to deal with computing time and space problem.

In this paper, we propose the Weighted Semantic PageRank (WSPR) algorithm, which uses semantic links directly for a more accurate page ranking. We utilize RDF [15] metadata to create a semantic-link-based Web structure from a hyperlink-based Web structure, and utilize this semantic information as inputs for WSPR. Using semantic links in a semantic-link-based Web structure helps resolve the problem in determining the meaning of the links provided by a hyperlink-based Web structure. We can compute the rank scores in a semantic-link-based Web structure by evaluating the meaning of links directly. Furthermore, WSPR is able to reduce the possibility of ranking less important pages with high scores, as it uses RDF resources on the pages to compute their ranks, thus giving pages with less important resources a smaller ranking score. In addition, we implemented WSPR algorithm using the MapReduce framework [1], which is a more effective parallel distributed processing method for analyzing a large volume of Web data.

The contributions of this paper can be summarized as follows:

- We propose a ranking algorithm that computes the importance of pages more accurately. When the algorithm evaluates the weight of links, it considers the semantics of the link directly, and does

not simply consider the number of links or use additional factors to estimate the meaning of links.

- The proposed algorithm prevents meaningless pages with many in-links to be mistaken for important pages. Resources, the semantic units of RDF instances, are used for computing the importance value instead of pages. Therefore, once a page receives a high importance value, the proposed algorithm guarantees that the page contains meaningful resources indicating important information.

- We developed WSPR system using MapReduce on Hadoop. This system has more computation capability for processing Big Data resources, enabling the proposed ranking algorithm to be utilized on the Web.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of PageRank and Extended PageRank algorithms, focusing on the evaluation of the link weights. In Section 3, we introduce a semantic-link-based Web structure. In Section 4, we present our WSPR algorithm using the MapReduce framework in detail. Section 5 reports the results of our experiments used to evaluate the validity of our proposal. Finally, in Section 6, we offer some concluding remarks regarding the proposed research as well as some directions for future work.

2 Related Work

A page-ranking algorithm is an essential Web information retrieval method as the volume of results matched with a given query during a retrieval step is hard to be managed by users. Such an algorithm answers a given user query with a page list ranked by importance. The early page-ranking algorithm was a term-based ranking algorithm, whose criterion for evaluating the importance of pages is how many matched terms [3, 4] are contained on the page. After 1998, alternative ranking algorithms based on a linked relationship of pages [5, 11] were provided, and proved that link-based ranking algorithms perform better than term-based algorithms.

$$PR(r_i) = d \sum_{j \to i} \frac{1}{N_j} \cdot PR(r_j) + (1 - d) \qquad (1)$$

where d is a damping factor, which can be set between 0 and 1. This damping factor is used to resolve the rank sink problem caused by a cyclic linked or non-linked Web structure. The damping factor is usually set to 0.85. In PageRank, the PageRank value of a page is the sum of the PageRank values of pages that refer to this page. Each page equally distributes its PageRank value to pages it links to.

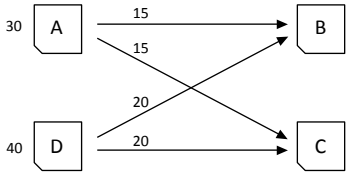

Fig. 1. A PageRank example.

In Figure 1, for example, page A, whose PageRank value is 30, assigns a PageRank of 15 to each of B and C. Similarly, page D assigns a PageRank of 20 to each of B and C. However, owing to its counting method, meaningless pages may be ranked highly by PageRank, which does not consider the meaning of links but only their number.

Weighted PageRank [26] is an alternative approach for avoiding a uniform distribution of rank values without proper consideration of the meaning of each linked relationship. Weighted PageRank evaluates the weight of the links to stratify the distribution of rank values (Figure 2). It computes the link weights using the proportions of in-links and out-links (Equation 2). However, this method is also based on the numbers of links, not their meaning. Furthermore, because a unit of ranking is page, it still exists for meaningless pages to be provided as important pages. Furthermore, because the method just estimates the importance of pages by their links, it does not always guarantee that a page actually contains important information.

$$W^{in}_{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p} \quad , \quad W^{out}_{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p} \qquad (2)$$

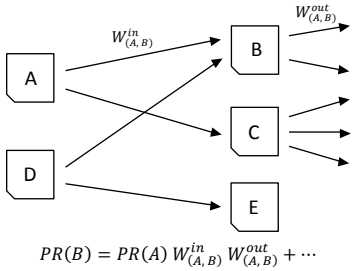

Fig. 2. A Weighted PageRank example.

Weighted Page Content Rank [18] improves Weighted PageRank by adopting Web content mining, through which it not only computes the link weights, but also observes the correlation between a given query and the resulting pages. However, this method still computes the weight of the links based on their number, and requires an extra cost involved with the mining process. Other methods such as Topic-Sensitive PageRank [27] and personalized PageRank [28] compute page importance using query-biased and user-biased metric. But our purpose is to generate an integrated page ranking algorithm as well as analyze semantic Big Data. Thus we set the scope of our research focused on unbiased page importance evaluation algorithm.

3 Semantic-link-based Web Structure

Semantic markup languages have been developed for better processing of Web information. Three representative semantic markup languages are RDFa [16], Microformats [14], and Microdata [13]. In this paper, we mainly focus on RDFa when building a semantic-link-based Web structure. RDFa was published in 2004 and received W3C recommendation in 2008. RDFa is a method used to describe RDF notations in XHTML (Figure 3). Web documents with RDFa can be read by Web browsers and extracted to obtain semantic information through RDFa parsers. The extracted information is a form of RDF [15]

metadata. RDF, which is a data model used to describe a set of knowledge, uses "triples" to express semantic relations among the knowledge set. A triple is composed of a subject, a predicate, and an object. This triple structure can be regarded as the unit of a graph dataset (Figure 4). Similar to RDF, RDFa is also a graph data model, and is more manageable for ranking algorithms than other semantic markup languages. Furthermore, this graph data model has an RDF predicate as a semantically labeled link, thus allowing the ranking algorithm to evaluate the weight of the links directly through their meaning.

Fig. 3. An example RDFa annotation.

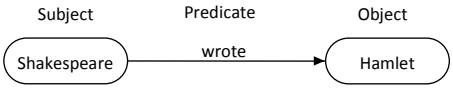

Fig. 4. An example RDF triple.

Several major sites have adopted RDFa which may help drive us toward the realization of the Semantic Web; Yahoo! and Google use RDFa for customizing their search results [21]; Facebook uses it for handling social data [22]; and Content Management Systems like Drupal and Wordpress, for semantic tagging [6]. Utilization methods of RDFa have also been provided; W3C has provided an RDFa distiller and a parser. In addition, RDFauthor [24] has provided an integrative approach for the management of RDFa data, and annotation systems [25, 7, 17, 10] have also been provided for various research fields.

Accordingly, it is clear that each of these methods is driving us closer to the existence of the Semantic Web. Thus, we consider the situation that pages contain semantic metadata using RDFa. If pages do not use RDFa notation for semantic metadata definition, we assume that these pages use other annotation method and use Information Extraction method to extract RDF format data.

4 Weighted Semantic PageRank

4.1 Proposed Architecture

Weighted Semantic PageRank (WSPR) system provides a new evaluation method that uses a semantic-link-based Web structure. It computes the weight of the links by evaluating their meaning directly. Four steps are used in this system (Figure 5). The first two steps change the environment from a hyperlink-based Web structure to a semantic-link-based Web structure. The other steps compute ranking values based on the structure constructed in the first two steps.

4.1.1 Semantic Information Extraction

As the first step of the WSPR algorithm, the system collects semantic information from the pages. Extracting semantic information is now much easier than before owing to our construction of a semantic-labeled Web structure described in the previous section. The system begins crawling through the Web using a parsing RDFa syntax. Figure 6 shows an example of RDF data extraction from a Web page containing an RDFa annotation. The WSPR algorithm uses a resource such as an object or a subject in an RDF triple as the unit of ranking. In other words, resources themselves are ranked, and the predicates are labeled links between resources.

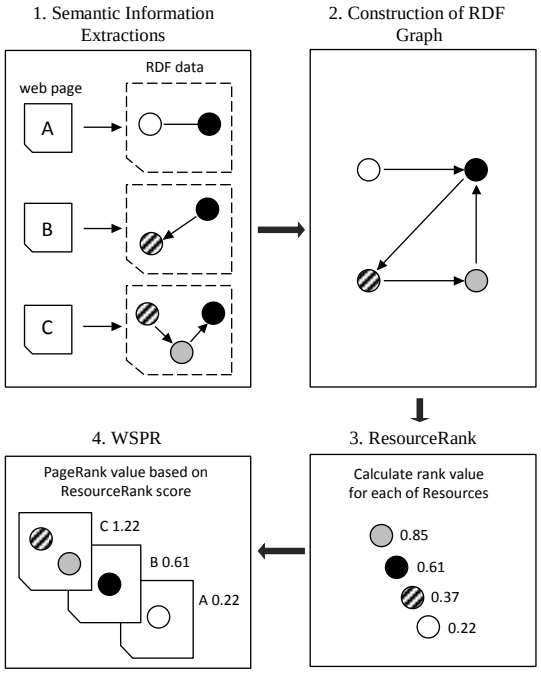

Fig. 5. Overview of the steps followed in the WSPR system.

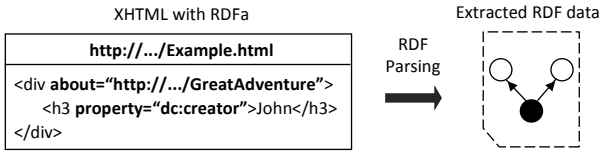

Fig. 6. RDF parsing of a Web page with an RDFa annotation.

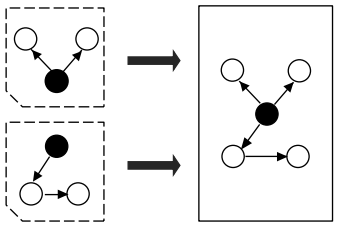

Fig. 7. Merging RDF triples with resources having the same URI.

4.1.2 Construction of RDF Graph

After the first step, the system then obtains a set of RDF data from the pages. A single series of RDF triples is insufficient for determining the rank value efficiently, and therefore multiple series must be interconnected together. Hence, this step uses a Uniform Resource Identifier (URI) as a key to find matched resources. For instance, supposing there are two triples to merge, as shown in Figure 7, the system checks the URI of their resources. When the system identifies that both triples have a resource with the same URI (the black nodes in Figure 7), the system merges the two triples into a single directed graph. In this way, this system creates a combined graph by merging all of the triples.

4.1.3 ResourceRank

In the third step, our system begins a ranking process called ResourceRank, which computes the ranking scores of the resources on the RDF graph built in the second step. ResourceRank can evaluate the weight of the links based on their own meaning since predicates labeled to contain semantic information are used to link the resources. This stratifies the distribution of rank values between linked resources based on the degree of their semantic relationships.

There are two types of methods, manual and automatic, used to evaluate the weight of the links [23]. We adopted the TF-IDF method to compute the weight of the links automatically. More specifically, the WSPR algorithm evaluates predicates instead of terms since it runs on an RDF graph. Therefore, it computes the Predicate Frequency (PF) as a Term Frequency. PF uses a function f which returns raw frequency of a predicate, and for normalization, the frequency divided by the maximum raw frequency of any predicate of the resource. IDF is also computed using predicates. The equations are as follows:

$$PF(p,r) = \frac{f(p,r)}{\max\{f(w,r): w \in r\}} \tag{3}$$

$$IDF(p,R) = log\frac{|R|}{|\{r \in R: p \in r\}|} \tag{4}$$

where p is a target predicate to compute the weight, r is a resource, and R is a set of resources.

Using PF-IDF, the value of a link weight is defined by using Equation 5.

$$weight(r_i,p) = PF(r_i,p) \times IDF(r_i,p) \tag{5}$$

Finally, ResourceRank equation takes on the form,

$$RR(r_i) = d \sum_{j \in outlink(i)} \frac{RR(r_j) \cdot weight(r_j,p)}{\sum_{j \in outlink(i)} weight(r_j,p)} + (1-d) \tag{6}$$

where RR(r_j) is the ResourRank value of a resource linked to resource ri, and is stratified based on its importance (weight) before being added to RR(r_i).

4.1.4 Weighted Semantic PageRank

The final step of this system is computing the PageRank value. In this step, the rank values of the pages are evaluated using the resource rank values calculated in the previous step. All resources originally contained on each page are in RDFa syntax forms. This means that the importance of a resource can be used to project the importance of the pages that contain this resource. That is, page importance is based on how many important resources, not how many in-links, a page has, unlike in previous ranking algorithms, which define page importance based on the latter criterion. This feature requires an important page with a greater PageRank value to contain important resources with meaningful information, and thus the probability that meaningless pages will be highly ranked is lower than in previous ranking algorithms.

Equation 7 shows the PageRank value using the ResourceRank values calculated in the previous step.

$$PageRank(p_i) = \sum_{r \in p_i} RR(r) \tag{7}$$

where RR(r) is the ResourceRank value of resource r, which is contained in page p_i. Thus, the PageRank value of page p_i is the summation of all ResourceRank values of the page pi resources.

4.2 MapReduce Algorithm

MapReduce methodology makes development of distributed and parallel processing more efficient. Google generated MapReduce framework, and Apache released Hadoop [2] - an open source implementation of Google's MapReduce framework. Hadoop has been extensively used on Big Data processing. On the MapReduce framework, researchers are able to concentrate on solving their own problems without having to manage distributed and parallel system directly.

A MapReduce job consists of map and reduce phases (Figure 8). In the map phase, input data is converted into key-value pairs. The key-value pairs are sent to the reduce phase by keys. In the reduce phase, data sets combined by key are processed for a specific purpose.

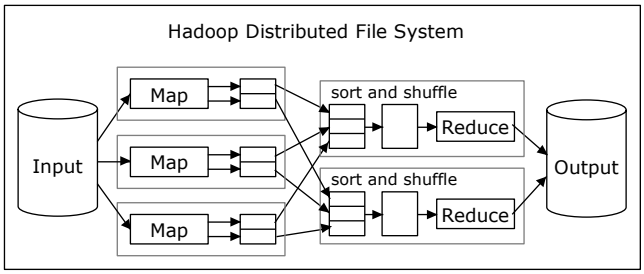

Fig 8. Overview of Hadoop MapReduce.

We implemented MapReduce version of WSPR in order to analyze large-scale semantic metadata. The WSPR MapReduce algorithm processes three jobs (Figure 9). The first job receives page information and their RDF metadata. The first job computes ResourceRank for each RDF resource until convergence, and the results of the first job are passed to the next job (Figure 10).

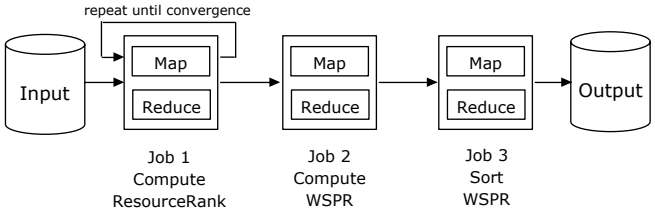

Fig. 9. WSPR MapReduce job framework.

```
class MAPPER
  method MAP(pageid i, page P)
    EMIT(pageid i, page P)   // Emit adjacency list
    for all pageid j ∈ P.AdjacencyList do
      r ← j.ResourceRank × j.LinkWeight
      EMIT(pageid j, r)        // Emit value for ResourceRank
    end

class REDUCER
  method REDUCE(pageid i, values [v1, v2, …])
    R ← ∅
    sum ← 0
    for all v ∈ values [v1, v2, …] do
      if IsResourceRankScore(v) then
        sum ← sum + v        // Sum of values for ResourceRank
      else
        R.AdjacencyList ← v   // Get adjacency list information
      end
    end
    R.ResourceRank ← sum × 0.85 + 0.15  // Compute rank
    EMIT(pageid i, page R)
```

Fig. 10. MapReduce Job 1: ResourceRank.

```
class MAPPER
  method MAP(pageid i, page P)
    EMIT(pageid i, P.resourceRank)

class REDUCER
  method REDUCE(pageid i, resourceRanks [r1, r2, …])
    R ← ∅
    sum ← 0
    for all r ∈ resourceRanks [r1, r2, …] do
      sum ← sum + r        // ResourceRank value summation
    end
    R.PageRank ← sum
    EMIT(pageid i, page R)
```

Fig. 11. MapReduce Job 2: WSPR.

```
class MAPPER
  method MAP(pageid i, page P)
    EMIT(P.PageRank, pageid i)  // Sort using Reduce function
```

Fig. 12. MapReduce Job 3: Ordering page by rank score.

In the second job, RDF resource information with ResourceRank score is used for computing WSPR score. RDF resource and RDF ResourceRank score pairs are grouped into pages each resource belongs to. WSPR score of each page is computed by summing up the group of ResourceRank scores assigned to each page (Figure 11).

The third job takes intermediate ranking information from the previous job as input data. Finally, the third job sorts pages by WSPR score and outputs the ranking result (Figure 12).

5 Experimental Evaluation

5.1 The Setup

The physical Hadoop cluster for the experiments comprises one master node and eleven slave nodes. Each node has 3.1 GHz quad-core CPU, 4GB memory, and 2TB hard disk. The operating system is 32-bit Ubuntu 12.04.2, the java version is 1.6.0_26, and the Hadoop version is 1.2.1.

As a source of Web data, we used 80,000 WikiPedia [9] web pages and extracted 500,000 RDF metadata from infobox tables in the WikiPedia pages.

5.2 Results

We evaluate WSPR and other systems according to precision, recall and f-measure. In Figure 13, the solid line indicates the results of WSPR, the dashed line with filled triangle is those of Weighted PageRank (WPR), and dashed line with cross is PageRank (PR) result. The result shows that WSPR has higher evaluation values than the others. This means WSPR provides less false positive and false negative ranking results. Similar conclusion can be drawn from Table 1, which shows the comparison among NDCG [29] of PR, WPR, and WSPR. We see that the results of WSPR attain higher values than those of the others.

Table 1. NDCG@k results for the test query

NDCG@k	PR	WPR	WSPR
NDCG@5	0.8765	0.9838	0.9931
NDCG@8	0.8824	0.9469	0.9748
NDCG@10	0.8866	0.9389	0.9732

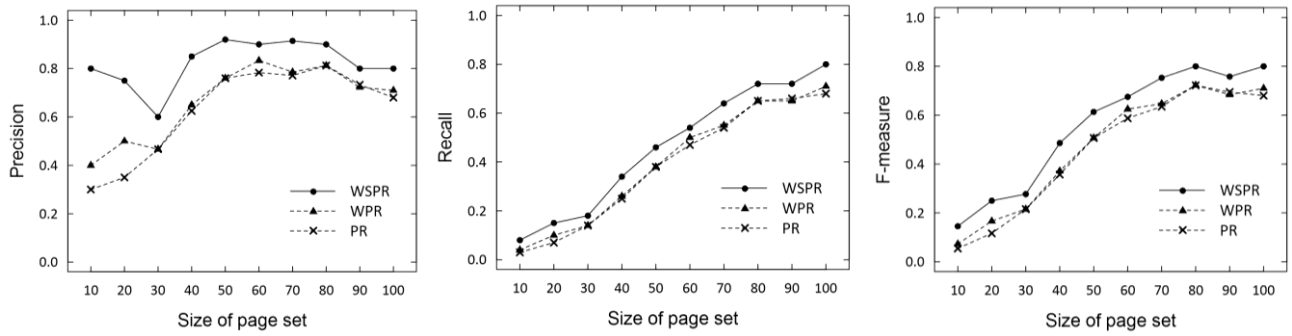

Fig. 13. Precision, Recall, and F-measure of PR, WPR, and WSPR for varying number of pages.

Table 2 shows a more detailed view of query results on literatures. In the ResourceRank stage, the third step of WSPR, the page on Macmillan has two resources: "Macmillan" and "Publishing company". The ResourceRank scores of these two resources are 1.118 and 0.429, respectively. On the other hand, the page on the United States has one resource, "United State," the ResourceRank score of which is 1.272. Although "United State" has the highest ResourceRank value among the three resources, the page on Macmillan has a higher WSPR score than the page of the United States (Table 3).

It is natural for human to choose the page on Macmillan as the most related page to the given query, since Macmillan is a publishing company, while the page on the United States appears irrelevant to be chosen as a related page. This shows that the result of WSPR takes semantic meanings of the pages into account.

Table 2. ResourceRank related within pages

RDF Resource	ResourceRank Score
"United State"	1.272
"Macmillan"	1.118
"Publishing company"	0.429

Table 3. Summary of ResourceRank used to compute WSPR

Page	RDF Resource (ResourceRank Score)	WSPR Score
Macmillan	"Publishing company" (0.429) "Macmillan" (1.118)	1.547
United States	"United State" (1.272)	1.272

Next we measured the processing time of semantic Big Data analysis. Figure 14 shows the execution time for the experiments. Each result with different data size supports linear growth in processing time rather than exponential growth. Thus, the results indicate that WSPR implemented using the MapReduce framework has two benefits in processing Big Data. First, it enables computation of large-scale semantic data. Second, the computation of the data takes relatively small amount of time compare to the other algorithms.

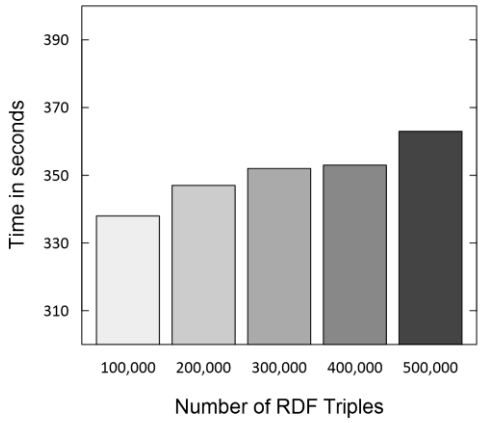

Fig. 14. MapReduce execution time

6 Conclusions

An RDF model can define concepts through the use of triples, which have a semantic link structure. In this paper, we utilized this feature to resolve a problem with PageRank in which the meaning of links used to compute importance cannot be properly evaluated.

Using a new ranking method that can be used to evaluate importance based on how many important resources Web pages have, WSPR provides more semantically relevant ranking results than other ranking methods. Therefore, once a page is ranked highly by WSPR, the page is guaranteed to contain important information related to the given query as WSPR ranks pages based on how many important resources, not how many in-links, the pages have, avoiding meaningless pages to be scored highly, which is a problem with other ranking methods.

Furthermore, we have adopted MapReduce framework to compute WSPR. Performance evaluations show that parallel and distributed processing on Hadoop is an effective way for semantic Big Data analysis.

Further research will be conducted using an automatic RDFa annotator. This will enable the WSPR algorithm to use both Web pages without semantic metadata and semantically annotated Web pages for computing a semantic rank score. We expect this to improve the adoptability of WSPR across the World Wide Web.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (No. 20110030812), by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (No. 20110017480), and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014R1A1A1002236).

References

[1] J. Dean and S. Ghemawat. “Mapreduce: Simplified data processing on large clusters,” In Proceedings of the 6th USENIX Symposium on Opearting Systems Design & Implementation (OSDI), pp. 137-150, 2004.

[2] Hadoop, Available from: http://hadoop.apache.org [Accessed: 12 March 2014].

[3] R. J. Bayardo, Y. Ma, and R. Srikant, “Scaling up all Pairs Similarity Search.” In Proceedings of the 16th international conference on World Wide Web, pp. 131-140, 2007.

[4] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing and Management, Vol. 24, No. 5, pp. 513-523, 1988.

[5] S. Brin and L. Page, “The Anatomy of a Large-scale Hypertextual Web Search Engine,” Computer Networks and ISDN Systems, Vol. 30, No. 1-7, pp. 107-117, 1998.

[6] S. Corlosquet, R. Cyganiak, A. Polleres, and S. Decker, “RDFa in Drupal: Bringing cheese to the web of data,” In Proceedings of the 5th Workshop on Scripting and Development for the Semantic Web at ESWC, 2009.

[7] M. Duma, “RDFa Editor for Ontological Annotation,” In Proceedings of the Student Research Workshop associated with RANLP, pp. 54-59, 2011.

[8] V. N. Gudivada, V. V. Raghavan, W. I. Grosky, and R. Kasanagottu, R, “Information Retrieval on the World Wide Web,” IEEE Internet Computing, Vol. 1, No. 5, pp. 58-68, 1997.

[9] Wikipedia, Available from: http://en.wikipedia.org/ [Accessed: 12 March 2014].

[10] A. Khalili, S. Auer, and D. Hladky, “The RDFa Content Editor - From WYSIWYG to WYSIWYM,” Proceedings of the IEEE Signature Conference on Computers, Software, and Applications, COMPSAC, 2012.

[11] J. M. Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” Journal of the ACM, Vol. 46, No. 5, pp. 604-632, 1999.

[12] M. Kobayasha and K. Takeda, “ Information Retrieval on the Web,” ACM Computing Surveys, Vol. 32, No. 2, pp. 144-173, 2000.

[13] W3C Working Group, "HTML Microdata," Available from: http://www.w3.org/TR/2011/WD-microdata-20110405/ [Accessed: 12 March 2014].

[14] R. Khare, “Microformats: The Next (Small) Thing on the Semantic Web?,” Journal IEEE Internet Computing archive, Vol. 10, No. 1, pp. 68-75, 2006.

[15] RDF Working Group, “Resource Description Framework,” Available from: http://www.w3.org/RDF/ [Accessed: 12 March 2014].

[16] W3C Working Group, "RDFa Core 1.1 - Second Edition," Available from: http://www.w3.org/TR/rdfa-syntax/ [Accessed: 12 March 2014].

[17] M. Samwald, E. Lim, P. Masiar, L. Marenco, H. Chen, T. Morse, P. Mutalik, G. Shepherd, P. Miller, and K. Cheung, ”Entrez Neuron RDFa: A Pragmatic Semantic Web Application for Data Integration in Neuroscience Research,” In Proceedings of the International Conference of the European Federation for Medical Informatics, pp. 317-321 , 2009.

[18] P. Sharma, D. Tyagi, and P. Bhadana, “Weighted Page Content Rank for Ordering Web Search Result,” International Journal of Engineering Science and Technology, Vol. 2, No. 12, pp. 7301-7310, 2010.

[19] A. Singhal, “Modern information retrieval: A brief overview,” Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 24, No. 4, pp. 35-43, 2001.

[20] K. Stein and C. Hess, “Information retrieval in trust-enhanced document networks,” In Proceedings of European Web Mining Forum (EMWF) 2005, and Knowledge Discovery and Ontologies (KDO) 2005.

[21] T. Steiner, R. Troncy, and M. Hausenblas, “ How Google is using Linked Data Today and Vision For Tomorrow,” Future Internet Assembly, Ghent, Belgium, 2010.

[22] The Open Graph Protocol, Available from: http://ogp.me [Accessed: 12 March 2014].

[23] N. Toupikov, J. Umbrich, R. Delbru, M. Hausenblas, and G. Tummarello, “DING! Dataset Ranking Using Formal Descriptions” In Proceedings of WWW 2009 Workshop on Linked Data on the Web (LDOW 2009), Madrid, Spain, 2009.

[24] S. Tramp, N. Heino, S. Auer, and P. Frischmuth, “RDFauthor: Employing RDFa for collaborative knowledge engineering,” In Proceedings of Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, Vol. 6317, pp. 90-104. Springer, Heidelberg , 2010.

[25] R. D. Virgilio, F. Frasincar, W. Hop, and S. Lachner, “A Reverse Engineering Approach for Automatic Annotation of Web Pages,” Multimedia Tools and Applications, Published online, 2011.

[26] W. Xing and A. Ghorbani, “Weighted PageRank Algorithm,” In Proceedings of Second Annual Conference on Communication Networks and Services Research CNSR 2004, Fredericton, N.B., Canada, pp. 305-314, 2004.

[27] T. H. Haveliwala, "Topic-sensitive PageRank," In Proceedings of the 11th international conference on World Wide Web, pp. 517-526, 2002.

[28] G. Jeh and J. Widom, "Scaling personalized web search," In Proceedings of the 12th international conference on World Wide Web, pp. 271-279, 2003.

[29] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” ACM Transactions on Information Systems, Vol. 20, No. 4, pp. 422-446, 2002.