International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 5, October 2016, pp. 2454~2461
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i5.10853

An Approach of Semantic Similarity Measure between Documents Based on Big Data

Mohammed Erritali*, Abderrahim Beni-Hssane, Marouane Birjali**, Youness Madani***
*TIAD Laboratory, Department of Computer Sciences, University of Sultan MoulaySlimane, Faculty of Sciences and Technologies, BéniMellal, Morocco
** LAROSERI Laboratory, Department of Computer Sciences, University of ChouaibDoukkali, Faculty of Sciences, El Jadida, Morocco

Article Info

Article history:

Received Apr 13, 2016
Revised Jun 29, 2016
Accepted Aug 11, 2016

Keyword:

Big Data
Document Similarity
Hadoop cluster
MapReduc programming model
Semantic Measure
WordNet

ABSTRACT

Semantic indexing and document similarity is an important information retrieval system problem in Big Data with broad applications. In this paper, we investigate MapReduce programming model as a specific framework for managing distributed processing in a large of amount documents. Then we study the state of the art of different approaches for computing the similarity of documents. Finally, we propose our approach of semantic similarity measures using WordNet as an external network semantic resource. For evaluation, we compare the proposed approach with other approaches previously presented by using our new MapReduce algorithm. Experimental results review that our proposed approach outperforms the state of the art ones on running time performance and increases the measurement of semantic similarity.

Corresponding Author:

MOHAMMED ERRITALI ,
TIAD Laboratory, Faculty of Sciences and Technologies, Department of Computer Sciences, University of Sultan Moulay Slimane
Béni Mellal, Morocco
Email: m.erritali@usms.ma

1. INTRODUCTION

Since the rise of the computer science, the volume of textual information stored continues to increase due to development of information technologies. These new technologies have enabled an exponential increase in the volume of data by online contents like blogs, posts, social networking and site interactions. Every day, 2.5 trillion bytes of data are created based on an estimate and it is very large amount so that 90% of data in the world was created in last 2 years [1]. This rapid increase in the volume of textual information has created the problem of how to find the relevant documents that interests us in this amount mass of textual information. To overcome this problem a discipline as a whole is born. This discipline is called the Information Retrieval (IR) of documents in a Big Data environment.

With the incessant increase of these documents, it has become difficult to manage and exploit them. This difficulty is closely related to the semantic aspect of these documents. Indeed, manual operation of is possible and gives good results. However, a manual procedure is not possible with large corpus. There are many applications using similarity detecting technology, Such as similarity recommendation [2], copy detection [3], social network mining [4] and so on. How to quickly detect similar documents becomes a basic and important problem as times go on. Document similarity computation is an important research topic in information retrieval and it is a key issue for automatic document categorization and clustering analysis. At present, it aims mainly to improve the accuracy and the efficiency with approaches such as the method based on vector space model [5], the method based on Map-Reduce model [6]. In the context of our work, we need

to design a new parallel algorithm based on MapReduce programming model to improve the value of the semantic similarity using WordNet and the running time performance.

In this paper, we are basically interested in the phase of the documents indexation, each document is represented by an intermediate representation. This representation is directly operated by the Information Retrieval System (IRS). It describes the contents of the document by descriptors. These descriptors are significant units in the document. In our context, to find the relevant documents by comparison with a document query, the ISR compares the representation of this query to the representation of each document. This comparison is done by means of a function of correspondence (Retrieval Status Value: RSV) and a score of relevance is assigned to each document. In most of the indexing process a weight assigned to each descriptor. This weight determines the discriminating power of the descriptor in the document where it is present. The majority of the approaches of the information retrieval exist in the literature takes only a simple words and/or fragments of the words for the research of documents and is unaware of the essential idea that takes the semantic relations of the words. The identification of the similarity between documents resulting from the indexing and the concepts of the semantic measure that is a fundamental phase in our work.

Our contribution of this research paper is to index the requestI_q, index each document (I_d) and compare the performance of the application to the representation of each document (RSV). This is formally translated as:

$$\begin{aligned} &I_q: Q \rightarrow E \\ &q \rightarrow I_q(q) \end{aligned} \quad (1)$$

$$\begin{aligned} &I_d : D \rightarrow E \\ &d \rightarrow I_d(d) \end{aligned} \quad (2)$$

$$\begin{aligned} &RSV : E \times E \rightarrow R^+ \\ &\left(I_q(q), I_d(d)\right) \\ &\rightarrow RSV\left(I_q(q), I_d(d)\right) \end{aligned} \quad (3)$$

Q is the set of queries,
D is the set of documents,
E is the set of descriptors.

The remainder of this paper is structured as follows:
In the next section, we briefly present the design and implementation of MapReduce programming model as a framework for managing distributed processing in a large of amount documents. We then study in section 4 the state of the art of the measures of similarity for comparing these measurements with our approach proposed in section 5. We conclude this section by presenting our proposed algorithm based on MapReduce model using the vector presentation of the documents and one of the approaches already existed in section 4. Finally, we conclude with the analysis and simulation of our approach, on Hadoop framework, before presenting the conclusion and the perspectives of this work.

2. RELATED WORK

Many studies have been presented on detecting document similarity in recent years for facilitating the search for information in complex information systems. Kumar et al. [7] and Chowdhury [8] surveyed duplicate or near duplicate data detection algorithms. Related work on text similarity detection can be mainly classified into two categories: traditional method and parallel method.

For the traditional methods, Lyon et al. [9] proposed a tri-gram and set theory-based algorithm, a data finger-based method, to extract the data finger of sentences and then mapped it into a range of value using Hash or MD5 function, then, reported the similarity according to the overlapped ratio of similar value or the maximum common sub-sequence. Matveeva [10] and Hatzivassiloglou et al. [11] presented a Vector Space Model (VSM) algorithm to compute the similarity using Cosine measurement of the vector. Yih [9] explored different score approaches, not traditional TF-IDF weight, to study the term weight function. Broder [12] explored a shingles-based algorithm to define the containment of two documents and took Jaccard coefficient [13] to represent the similarity of them.

For the parallel-based methods, most approaches focused on MapReduce model. Zhang et al. [14] presented a sequence-based method to detect partial similarity of web page using MapReduce, which consisted of two sub-tasks as sentence level near duplicate detection and sequence matching. In this work,

will be also used MapReduce framework, but we integrate it with some effective features to guarantee running time performance.

3. BIG DATA CHALLENGES BY HADOOP

Hadoop is an open source Apache software framework that evaluates gigabytes or petabytes of structured or unstructured data and transforms it more manageable for applications to work with this large data[15]-[16]. The core components of Hadoop are HDFS and MapReduce. HDFS is basically used to store large data sets and MapReduce is used to process such large data sets.

3.1. Hadoop Distributed File System (HDFS) architecture

The Hadoop Distributed File System (HDFS) is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. In a large cluster, thousands of servers both host directly attached storage and execute user application tasks [17]. HDFS uses a write-once, read-many model that breaks data into blocks that it spreads across many nodes for fault tolerance and high performance, as Figure 1.

HDFS stores file system metadata and application data separately on a dedicated server, called the NameNode. Application data are stored on other servers called DataNodes.

NameNode: the node that controls the HDFS. It is responsible for serving any component that needs access to files on the HDFS. It is also responsible for ensuring fault-tolerance on HDFS. Usually, fault-tolerance is achieved by replicating the files over different nodes.

DataNode: this node is part of HDFS and holds the files that are put on the HDFS. Usually these nodes also work as TaskTracker. JobTracker tries to allocate work to nodes such files accesses are local, as much as possible.

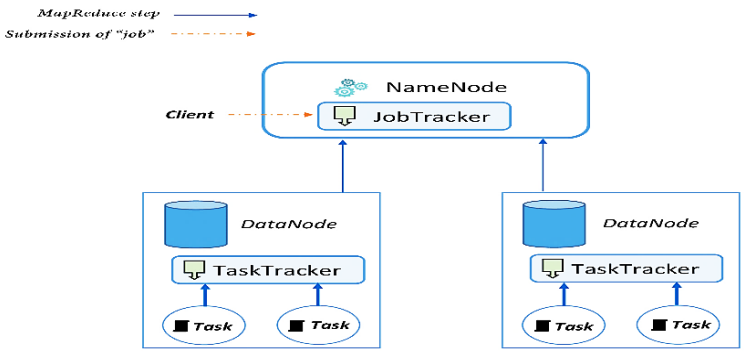

Figure 1. The interaction between HDFS and MapReduce job

At the level of the NameNode, the *JobTracker* is responsible for the management of resources that is the control of the DataNodes in cluster. It manages the entire duration of the life of a job. The *TaskTracker* has responsibilities more simple, namely launch the tasks in the order provided by the JobTracker and periodically give a status of progress of the task to the JobTracker.

3.2. MapReduce programming model

MapReduce is a programming model and an associated implementation for processing and managing large data sets with a parallel, distributed algorithm on a cluster. MapReduce divides into three parts: Map, Shuffle and Sort, and Reduce. A Map part of MapReduce job splits the input datasets into independent chunks. The independent chunks are processed in a completely parallel manner using Map task. Then the Reduce function merged these values to form a possibly smaller set of values. That is, the Reduce function filtered the Map output and produces the results with respect to the key of the Map phases [18].

$$\textbf{\textit{Map:}} \quad < \mathrm{doc_i}, \mathrm{docText} > \quad \rightarrow \quad < \mathrm{docID_i}, \mathrm{term_j} >$$

$$\textbf{\textit{Reduce:}} \quad \left.\begin{array}{l} < \mathrm{docID_i}, < \mathrm{term_j}, \mathrm{weight_j} >> \\ < docQuery, < \mathrm{term_i}, \mathrm{weight_i} >> \end{array}\right\} \rightarrow < \mathrm{docID_i}, \ \mathrm{Sim_i} >$$

Sim_i: is the semantic similarity between the document $doc_i$and the query $docQuery$.

In our proposed algorithm runs on two consecutive MapReduce phases, the first to build an indexing phase and the second to compute the semantic similarities measures, as our design MapReduce shown in the Figure 2.

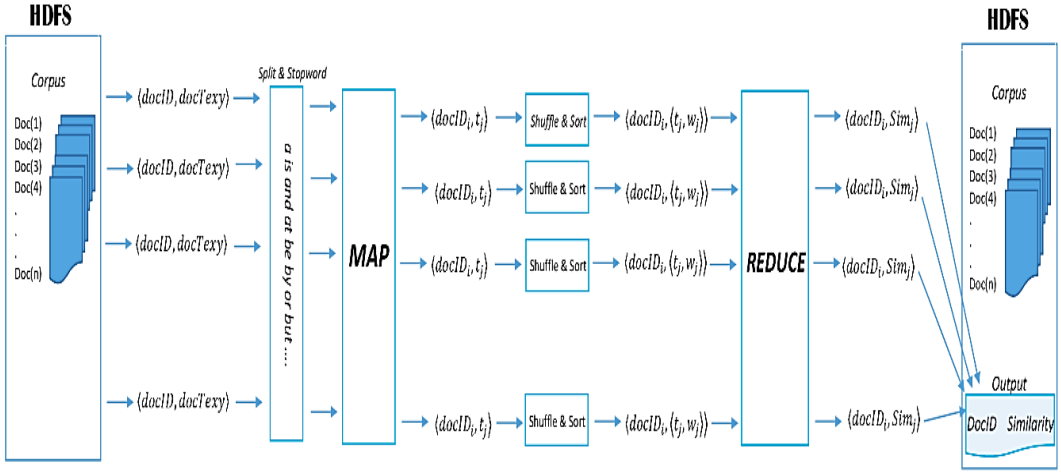

Figure 2. The process of our MapReduce model

Document indexing: given a corpus, for each term t_i of document, the mapper emits the document ID as the key, and his words as the value. The shuffle phase of MapReduce, groups these words by a collection of the values of each document, and delivers these inverted lists to the reducers, that write them to blocks.

Semantic similarities measures: In this step, Reduce takes the output of the Map function and computes the semantic relation between each collection of values of each document and the query. This semantic relation computed byWordNet as an externelsemantic network [19] with the use of the weight of the words w_i and one of the approaches already existed to compute the semantic similarity between the two concepts.

4. SEMANTIC SIMILARITY MEASURES

In this related work section, the contribution of our semantic similarity measure is to evaluate the semantic proximity between documents.We presentdifferent approaches of the similarity measure between words or documents. There are three main families approaches in the literature of semantic measurement between the documents, approaches based on the Arcs, approaches based on the Nodes and Hybrid approaches.

4.1. Approaches based on the Arcs

The majority of similarity measurement of concepts in ontology are based on their distances [20]. Obviously, the concept X is more similar to a concept Y than a concept Z, this similarity are evaluated by the distance, which separates the concepts in ontology. These measurements make use of the hierarchical structure of ontology to determine the semantic similarity between the concepts.

Rada et al. measure: This measure [21] is adopted in a network semantics and is based on the fact that we can compute the similarity based on the links hierarchical (generalization) "is-a". To compute the similarity of two concepts in an ontology, we must calculate the number of minimum Arcs which separate them. This measure, based on the computation of the distance between the nodes by the shortest path.
The similarity measure with this measurement between the concept $C_1$and the concept $C_2$is as well of the formula:

$$\mathrm{Sim}(c_1, c_2)_{\mathrm{Rada}} = \frac{1}{1 + \mathrm{dist}(c_1, c_2)} \quad (4)$$

*Wu and Palmer measure:*The principle of this measurement is given an ontology formed by a set of nodes and a root node (R). X and Y represent two ontology elements for which we will compute the similarity. The principle of similarity measurement is based on the distances (N1 and N2) which separate the X and Y nodes from the node R and the distance (N) which separates the Subsuming Concept (SC).

The Wu and Palmer measurement is defined by this formula:

$$\mathrm{Sim}(X,Y)_{\text{Wu and Palmer}} = \frac{2\times N}{N1+N2} \quad (5)$$

4.2. Approaches Based on the Nodes

These techniques adopt a new measure in terms of the entropic measurement $\boldsymbol{E}$ of the information theory [22].

$$E(c) = \log(P(c)) \quad (6)$$

Where $P(c)$is the probability of finding a value concept of$c$.

*Resnik measure:*The notion of the Informational Contents (IC) was initially introduced by [23], which proved that an object (word) is defined by the number of the specified classes and that the semantic similarity between two concepts is measured by the quantity of information which they share. The informational contents are obtained by computing the object frequency in the corpus.The formula of this measure is:

$$\mathrm{Sim}(c_1,c_2)_{\text{Resnik}} = \mathrm{Max}\,[CS(c_1,c_2)] = \mathrm{Max}[-\log P(CS(c_1,c_2))] \quad (7)$$

$CS(c_1,c_2)$: represents the most concept specific (which maximizes the similarity value) between the concept c_1 and $c_2$in the ontology.

Lin's Measure: Lin has defined a different similarity measure that of Resnik by this formula:

$$\mathrm{Sim}(c_1,c_2)_{\text{Lin}} = \frac{2\times \log(P(CS(c_1,c_2)))}{\log P(c_1)+\log P(c_2)} \quad (8)$$

4.3. Hybride Approaches

These techniques are founded on a model which combines between the approaches based on the Arcs in addition to the informational contents which are regarded as factor of decision.

Jiang and Conrath Measure: To cure the problem presented to the level of the Resnik measurement, Jiac [24] brought a new formula which consists in combining the Entropy (Informational Contents) of the specific concept to those of the concepts which we seeks the similarity. This approacheis computed by the formula following:

$$\mathrm{Sim}\,(X,Y)_{\text{Jiang}} = \frac{1}{\mathrm{distance}(X,Y)} \quad (9)$$

The distance between X and Y is computedby the following formula:

$$\mathrm{distance}(X,Y) = E(X)+E(Y)-\left(2\times E\big(CS(X,Y)\big)\right) \quad (10)$$

Leacock and Chodorow: Another method presented by [19], which combines between counting of the arcs method and the informational contents method. The proposed measure by Leacock and Chodorowis based over the shortest way length between two synsets of Wordnet. This technique is defined by the formula:

$$\mathrm{Sim}_{\,lc}(X,Y) = -\log\left(\frac{cd(X,Y)}{2\times M}\right) \quad (11)$$

$M$is the longest way length, which separates the concept root, of ontology, of the concept more in bottom. We indicate that $cd(X,Y)$ is the shortest way length which separates X of Y.

4.3. Approaches Based on the Vector Space

These approaches use a characteristic vector, in a dimensional space, to represent each object and calculate the similarity while being based at the Cosine measurement or the Euclidean distance. The similarity definition between two vectors of objects is obtained by their internal contents. Here are some approaches mentioned in the literature:

Jaccard Measure: It's defined by the common objects number divided by the objects full number minus the common objects number:

$$\text{Sim}_{\text{Jaccard}}(X,Y) = \frac{\sum_{i=1}^{n} x \times y}{\sqrt{(\sum_{i=1}^{n} x^2)} + \sqrt{(\sum_{i=1}^{n} y^2)} - \sum_{i=1}^{n} x \times y} \quad (12)$$

Cosine Measure: It uses the complete vector representation, that is to say the objects frequency (words). Two documents are similar if their vectors are combined. If two objects are not similar, their vectors form an angle (X, Y) whose Cosine represents the similarity value. The formula is defined by the ratio of the scalar product of vectors x and y and the product of the norm of x and y.

$$\text{Sim}_{\text{Cosine}}(X,Y) = \frac{\sum_{i=1}^{n} x \times y}{\sqrt{(\sum_{i=1}^{n} x^2)} \times \sqrt{(\sum_{i=1}^{n} y^2)}} \quad (13)$$

The measurement of Cosine quantifies the similarity between the two vectors as the cosine of the angle between two vectors.

Euclidean Measure: The Euclidean similarity is based on the ratio of the Euclidean distance increased by 1. The Euclidean distance is de denies by the following formula:

$$\text{dist}_{\text{Euclidean}} = \sqrt{\sum_{i=1}^{n} (x-y)^2} \quad (14)$$

The similarity measure is therefore defined by:

$$\text{Sim}_{\text{Euclidean}}(X,Y) = \frac{1}{1 + \text{dist}_{\text{Euclidean}}} \quad (15)$$

5. OUR PROPOSED APPROACH

In this section, we present a general and schematic view of the steps of our approach as well as their description, to achieve our process forcompute the semantic similarity measures between documents.

Our work raises a new approach tocompute the semantic similarity between documents by applying our hybrid approach based on the approaches already mentioned in the last section and the vector representation of documents.The general objective is to search in a large corpus stored in HDFS, the most relevant documents to a user request that is composed by a document.

The Figure 3 shows the steps to compute the similarity between a query and a document on applying our approach. Our approach implement an indexing step documents to present each document by the words that compose it. This step has much sub-steps namely: tokenization, stemming, elimination uppercase, stopwords ...Then, this vector representation was enriched with semantic networkbased on the synonyms of words using Wordnet.For each word of the document,which has associated synonyms in WordNet as synset and we added weight of these synonyms.The returned document is sorted by decreasing order of semantic similarity.

5.1. Our proposed MapReduce algorithm

In our proposed MapReduce algorithm runs on two consecutive MapReduce phases, the first to build an indexing phase and the second to compute the similarity measure.

Indexing of documents: given a corpus, for each term of document, the mapper emits the document ID as the key, and his words as the value.

Semantic similarity measure: In this step, Reduce will take the output of the Map function and makes the semantic indexation of all words using WordNet for computing the similarity relation between each collection of values of each document and the query. This similarity measure computed by our algorithm with the use of the weight of the words and one of the approaches already existed in the next section.

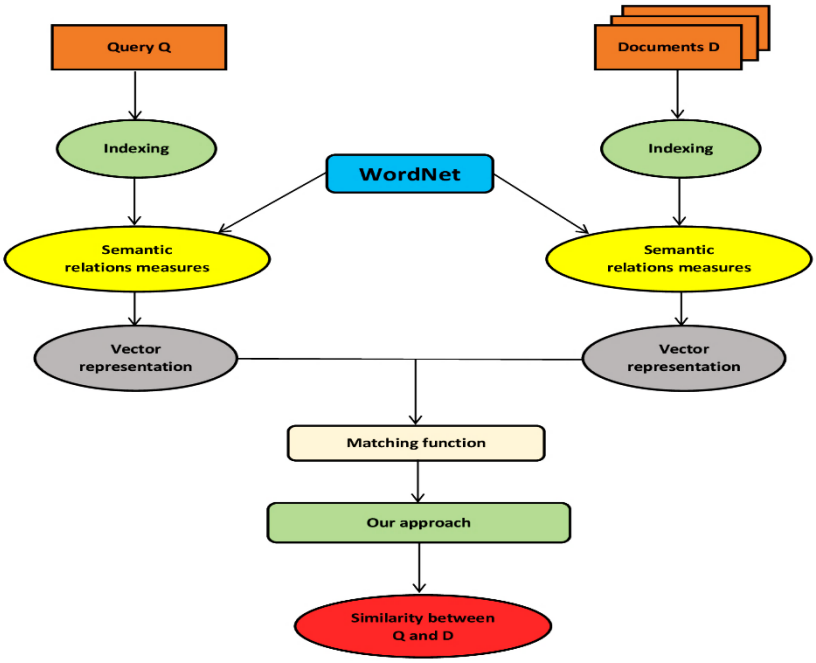

Figure 3. The process of our methodology to compute the semantic similarity measure

This work could be expressed using the followingMapReduce algorithm:

Class Mapper

Method **Map**(Docid,term)

This map method is called once per input line; map tasks are run in parallel over subsets of the input files.

RemoveStopWord&removePunctuation&ComposedWord (term)

The value contains an entire line from your file. We tokenize the line using StringUtils

For each element ∈ (Docid,term)

Write(Docid,term)

For each word the map outputs the word as the key and the document ID as the value.

End for

Class Reducer

Method **Reduce**(Docid,List(term))

The reduce method is called once per unique map output key. The Iterable allows to iterate over all the values that were emitted for the given key.

List(q) = indexing(Query)

Our semantic indexing method

S=0

X ← 0

Y ← 0

For each n ∈List(term)

We keep a set of all the document IDs that you encounter for the key.

F=calculateoccurence(n)

For each e ∈List(q)

We add the document ID to our set. The reason you create a new text object is that MapReduce reuses the text object when iterating over the values, which means we want to create a new copy.

R= calculateoccurence(e)

Count the number of occurrences for each term

X←X+F×R×Sim(n, e)

Y←Y+F×R

End for

End for

S←X/Y

Write(Docid,S)
Our reduce outputs the document IDs and the semantic similarity measure of each document.
End

Our approach is a hybrid, we use the vector presentation of the documents and one of the approachesalready exist in section 4. Our approach is presented by the following formula:

$$\mathrm{Sim}(q,d) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} q_i \times d_j \times \mathrm{Sim}(i,j)}{\sum_{i=1}^{n}\sum_{j=1}^{m} q_i \times d_j} \quad (16)$$

i: represents the concepts of the query q
j: represents the concepts of the document d
q_i: is the frequency of the concept i in query q
d_j : is the frequency of the concept j in document d
$Sim(i,j)$: is the semantic similarity between the two concepts i and j using WordNet.

6. RESULTS AND ANALYSIS

The choice of approach from the approaches presented in section 4 is very important in our proposed approach because it plays an important role in the research.To perform the comparison between these approaches, we compute the semantic similarity measure with the same of documents and choose the approach that gives good results. This table shows the computed similarity between the same documents.

Table 1. Comparison of running time and similarity measure of different approaches

Approaches	Similarity measure	running time (msec)
Leacock and Chodorow	0.14	1016
Wu and Palmer	0.11	1297
Resnik	0.07	1360
Jiang Conrath	0.04	1391
Lin	0.02	1344

The results show that the similarity changes with the change of each approach, it is very important in the Leacock and Chodorow approach because it gives the greatest similarity with a minimum running time.Based on this evaluation, our approachwill be basedon Leacock and Chodorow approach.

6.1.Evaluation of our approach

In our experiments,we used a physical machine that equipped by an Intel Core i3 CPU 2.27GHz, 8 GB of memory. For virtualization, Xen Server 6.2 was installedas a Hypervisor of type 1 based on the distribution provided by Citrix. Our machine is a virtualized Linux Ubuntu Server 12.04, with 6GB ofmemory for NameNode/DataNodes. During of our work we have used the version 5.1.2 of Cloudera Manager that unifies via a user graphical interface of the installation, configuration and management.

Figure 4 shows the results of the semantic similaritymesures between thesame documentusing our approach and the approachesalready exist in the literature.

Our approach provides the double of the semanticsimilarity measure compared to other approachesbecause it is based on the LearockandChodorow approach for the method Cosine andthe vector representation of the semantic relations between the the conceptswith the use of the weight of the words and WordNet.

Figure 5 shows the results by applying our approach to compute the semantic similarity between a document and a text corpus that contains contains a variable number of documents by computing in each case the time necessary to find the similarity between the query and each document of the corpus to return a list of documents that are semantically similar (relevant) to a user request.

Table 2 demonstrates the running time performance of ourMapReduce algorithm,it shows how our new algorithmprofit of the parallelizationof processing and managing the large number of documents with a parallel, distributed tasks on Hadoop.

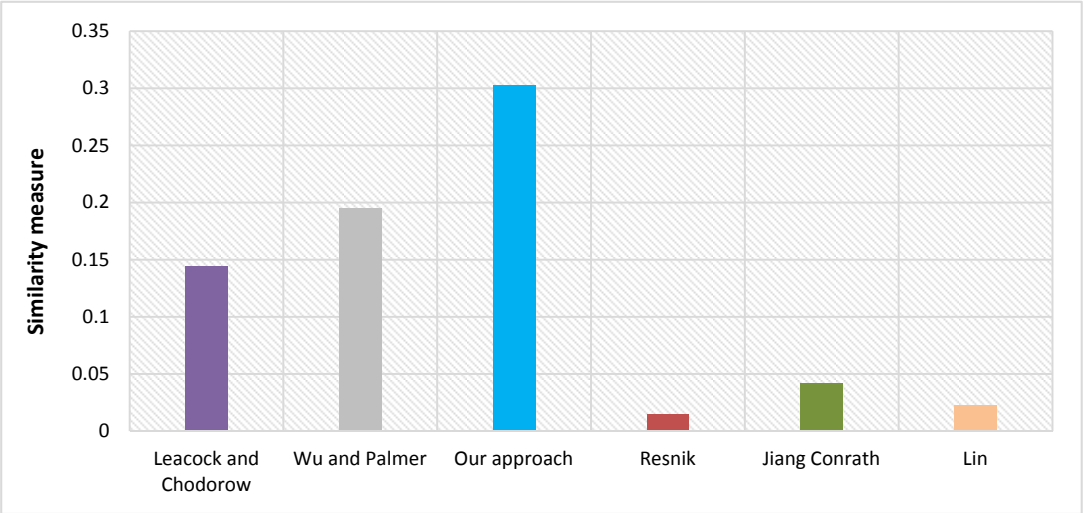

Figure 4. Comparison betweenthe semantic similarity measure of our approach and other approaches

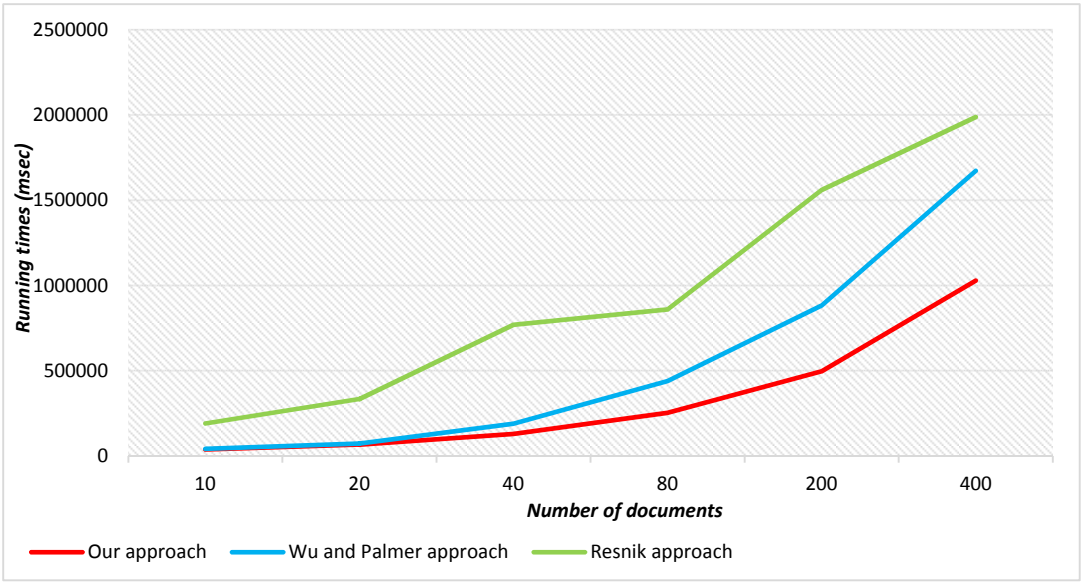

Figure 5. Comparison betweenperformance of running timesinour approach and other approaches

Table 2.Evaluation of running time for our MapReduce algorithm

Number of documents	Map	Reduce
10	1067	36341
20	2974	40636
60	4845	35791
100	16381	172799
200	32562	406338
400	10336	1068975

7. CONCLUSION

This paper discussed our approach based on a new MapReduce algorithm to compute the similarity between a query and documents existing in HDFS and find the most pertinent of documents.The results conclude that our Mapreduce algorithm outperforms the state of the art ones on running time performance and increases the measurement of semantic similarity.The future research involves testing multilingual of WordNet for documents with using Hadoop multi-nodes to improve these resultsand analyzing the Graphics Processing Unit (GPU). The work is underway and results will be available soon.

REFERENCES

[1] Improving Decision Making in the World of Big Data http://www.forbes.com/sites/christopherfrank/2012/03/25/improving-decision-making-in-the-world-of-big-data/#7a89869c4b4d

[2] Khadija A. Almohsen, Huda Al-Jobori, "Recommender Systems in Light of Big Data", International Journal of Electrical and Computer Engineering (IJECE), Vol. 5, No. 6, December 2015, pp. 1553-1563, 2015.

[3] Hoad T. C. and Zobel. J., "Methods for Identifying Versioned and Plagiarized Documents," in JASIST,vol. 54,pp. 203- 215, 2003.

[4] Spertus E.,Sahami M. and Buyukkokten O., "Evaluating Similarity Measures: A Large-scale Study in the Orkut Social Network," in proceedings of KDD, 2005.

[5] Mao, E., Wesley, P. and Chu, W. (2007) The Phrase Based Vector Space Model for Automatic Retrieval of FreeDocument Medical Documents. Data & Knowledge Engineering, 1.

[6] He, C.B., Tang, Y. and Tang, F.Y. (2011) Large-Scale Document Similarity Computation Based on Cloud Computing Platform. 2011 6th International Conference on Pervasive Computing and Applications (ICPCA).

[7] Chowdhury A., "Duplicate data detection," retrived from http://ir.iit.edu/~abdur/Research/Duplicate.html,2004.

[8] Lyon C., Malcolm J. and Dickerson B., "Detecting Short Passages of Similar Text in Large Document Collections," in processing of EMNLP,2001.

[9] Matveeva I., "Document Representation and Multilevel Measures of Document Similarity," in proceedings of ACLHLT,2006.

[10] Hatzivassiloglou V.,Klavans J.L. and Eskin E., "Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Cominations via Machine Learning," in proceedings of SIGDAT,1999.

[11] Yih W., "Learning Term-weighting Functions for Similarity Measures," in proceedings of EMNLP,2009.

[12] Broder A. Z., "On the Resemblance and Containment of Documents," in proceedings of SEQUENCES,1997.

[13] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn and Supachanun Wanapu, "Using of Jaccard Coefficient for Keywords Similarity", Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013, March 13 - 15, 2013.

[14] Qi Zhang,Yue Zhang,Hao. Yu and Xuan. Huang, "Efficient Partial-Duplicate Detection Based on Sequence Matching," in proceedings of SIGIR,2010.

[15] Hamid Bagheri and Abdusalam Abdullah Shaltooki, "Big Data: Challenges, Opportunities and Cloud Based Solutions", International Journal of Electrical and Computer Engineering (IJECE), Vol. 5, No. 2, pp. 340-343, April 2015.

[16] Apache Hadoop, http://hadoop.apache.org/

[17] Konstantin Shvachko, HairongKuang, Sanjay Radia, Robert Chansler, "The Hadoop Distributed File System".

[18] Yenumula B Reddy, "Document Identification with MapReduce Framework," DATA ANALYTICS: The Third International Conference on Data Analytics, 2014.

[19] François-Régis Chaumartin, "WordNet et son écosystème : un ensemble de ressources linguistiques de large couverture" : https://hal.archives-ouvertes.fr/hal-00611240.

[20] Lee J.H., M.H.Kim and Y.J.Lee, ''Information Retrieval Based on Conceptual Distance in IS A Hierarchy'', Journal of Documentation 49, pp. 188- 207, 1993.

[21] Rada R., Mili H., Bicknell E., &Blettner M. (1989). Development and application of a metric on semantic nets. IEEE Transaction on Systems, Man, and Cybernetics, 19(1):17-30.

[22] D. Lin. An Information-Theoretic Definition of similarity. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98). Morgan- Kaufmann: Madison, WI, 1998.

[23] Resnik P., Using information content to evaluate semantic similarity in taxonomy. In Proceedings of 14th International Joint Conference on Artificial Intelligence, Montreal, 1995.

[24] Jiang J. and D. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in Proceedings of International Conference on Research in Computational Linguistics, Taiwan, 1997.