Hindawi
Computational and Mathematical Methods in Medicine
Volume 2021, Article ID 7937573, 11 pages
https://doi.org/10.1155/2021/7937573

Research Article

An Efficient Parallelized Ontology Network-Based Semantic Similarity Measure for Big Biomedical Document Clustering

Meijing Li,[1] **Tianjie Chen**,[1] **Keun Ho Ryu**,[2,3,4] **and Cheng Hao Jin**[5]

[1]*College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China*
[2]*Data Science Laboratory, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh 700000, Vietnam*
[3]*Biomedical Engineering Institute, Chiang Mai University, Chiang Mai 50200, Thailand*
[4]*Department of Computer Science, College of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea*
[5]*ENN Research Institute of Digital Technology, Beijing 100096, China*

Correspondence should be addressed to Meijing Li; mjli@shmtu.edu.cn

Received 27 May 2021; Accepted 11 October 2021; Published 9 November 2021

Academic Editor: Lin Lu

Semantic mining is always a challenge for big biomedical text data. Ontology has been widely proved and used to extract semantic information. However, the process of ontology-based semantic similarity calculation is so complex that it cannot measure the similarity for big text data. To solve this problem, we propose a parallelized semantic similarity measurement method based on Hadoop MapReduce for big text data. At first, we preprocess and extract the semantic features from documents. Then, we calculate the document semantic similarity based on ontology network structure under MapReduce framework. Finally, based on the generated semantic document similarity, document clusters are generated via clustering algorithms. To validate the effectiveness, we use two kinds of open datasets. The experimental results show that the traditional methods can hardly work for more than ten thousand biomedical documents. The proposed method keeps efficient and accurate for big dataset and is of high parallelism and scalability.

1. Introduction

Recently, researchers pay much attention to semantic information discovery. Semantic data mining has been introduced into various fields of text mining, such as text clustering [1, 2], text classification [3, 4], information extraction [5–7], named entity recognition [8–10], and sentiment analysis [11–13]. Machine learning is the most commonly used method in text mining. In the latest research for text classification, ensemble strategy is often applied, which can capture multiple characteristics from complex text data [14–17].

For text clustering, with the continuous growth of data scale, it poses a challenge for people to mine information hidden in big text data. Since the similarities between texts are required before clustering, it is imperative to explore effective methods of computing similarity under the big data background [18].

Document clustering is an important application in the text clustering domain which helps people navigate the interested documents conveniently [19, 20]. Detecting the text similarity is of great importance in document clustering, which directly affects the performance of clustering. Numerous studies about similarity detection have been proposed, including vector-based [21–23] and ontology-based [24, 25]. The vector-based methods change the text into vector representation and then view the cosine similarity between vectors as the text similarity. The ontology-based methods use a structural knowledge representation network to describe the meanings and relationships of concepts. Since the vector-based methods ignore the semantic information between words, the ontology-based method attracts much attention at present [26].

An ontology is a hierarchical structure in which concepts are represented as nodes. And the nodes are connected with some relationships such as “is a” and “part of.” Thus, the

semantic similarity between concepts can be quantified in an ontology by detecting the node correlation in the structure. Existing ontology-based semantic similarity measurements can be divided into four categories. The first type is path-based, which takes the path distance between nodes in the structure as a measure of correlation. Bulskov et al. [27] used the path length between two nodes in the ontology. The path length is computed by the edges connecting the nodes. Wang [28] gave precomputed weights to edges on the basis of Buskov's method. The second type is information content (IC) based. Information content is the amount of information that a concept expresses, which can be computed from the ontology and corpus. The more a concept occurs, the less information content it has. Resnik [29] took the IC value of the least common ancestor (LCA) of the two nodes as the semantic similarity. Lin [30] extended the method by normalizing the IC value of the LCA using the IC value of both nodes. The third type is depth-based. Leacock and Chodorow [31] and Li et al. [32] took the depth of nodes in the ontology into account since the depth of nodes represents the information specificity. The fourth type is hybrid. Hybrid methods use more than one class of information. Jiang and Conrath [33] and Zhao and Wang [34] combined the IC and depth of nodes to compute the similarity.

In the domain of biomedical text mining, Medical Subject Headings (MeSH) is one of the most commonly used ontologies, which contains 29,638 MeSH headings arranged hierarchically in a tree structure by 2020 [35, 36].

Nowadays, with the rapid development of biomedicine, the amount of biomedical literature grows rapidly. Even if people narrow the search scope, a lot of literatures are retrieved. For instance, over the past five years, PubMed (http://pubmed.ncbi.nlm.nih.gov) has indexed more than 900 hundred biomedical citations by querying "cancer" in all fields. In addition, due to the complexity of ontology-based semantic similarity calculation, computing semantic similarity between a big number of documents leads to low efficiency. Our experiments show that the existing methods can hardly work with more than ten thousand documents. However, clustering is more valuable when the amount of data is larger.

To solve these problems, we proposed a method on the basis of Hadoop MapReduce. Hadoop is a framework that allows for distributed processing across clusters of computers. MapReduce is a module of Hadoop, allowing the parallel processing of large data sets. Traditionally, the document similarity is computed pair by pair, which causes redundant computation. The proposed method parallelizes the process of computing document similarity for the purpose of reducing the computational redundancy and increasing the amount of data that can be processed.

2. Materials and Methods

2.1. Definition. The set of documents to be clustered is denoted by $D(D = \{d_1, d_2, \cdots, d_n\})$. Similarly, the set of MeSH headings is denoted by $M(M = \{m_1, m_2, \cdots, m_n\})$. In this article, we define the MeSH headings as the semantic features of biomedical documents since the MeSH headings describe the subject of each article in MEDLINE. Thus, we use a set of MeSH headings to represent a document: $d = \{m_{k_1}, m_{k_2}, \cdots, m_{k_n}\}$, where j is the index of MeSH headings.

Table 1: An example of a biomedical document (PMID: 10496010) with corresponding MeSH headings and tree number of nodes.

MeSH heading	Tree number
DNA repair	G02.111.222
	G05.219
Genetic diseases, inborn	C16.320
Humans	B01.050.150.900
	.649.313.988.400
	.112.400.400

In the MeSH ontology, each MeSH heading is mapped to one or more nodes associated with tree numbers. The deeper the node is, the more specific the information is. The MeSH Tree nodes are denoted by $V(V = \{v_1, v_2, \cdots, v_n\})$. Similarly, a set of nodes are used to represent a MeSH heading: $m = \{v_{j_1}, v_{j_2}, \cdots, v_{j_n}\}$, where j is the index of nodes.

Table 1 shows an example of a document with MeSH headings and corresponding tree number of nodes.

Define $\mathrm{Sim}(\cdot, \cdot)$ a function that outputs the similarity between two inputs. For example, $\mathrm{Sim}(v, v')$ outputs the similarity between two nodes, and $\mathrm{Sim}(m, d)$ outputs the similarity of a MeSH heading to a document. Define $lca(v, v')$ outputs the LCA (least common ancestor) of two nodes in the MeSH ontology.

In MapReduce programming model, data is represented as key-value pairs. The key-value pair is denoted by $<k, v>$. Generally, a MapReduce task mainly consists of three stages: map, shuffle, and reduce. The input file is first divided into multiple splits through the input format, and each split will be assigned a map task. The map task processes the input file line by line and outputs intermediate key-value pairs: $<k_1, v_1>$. Shuffle is a process after the map task. Shuffle copies data from the map task to the reduce task, sorts the data according to the key value, and aggregates data with the same key: $<k_2, \mathrm{list}(v)>$. The reduce task processes the shuffled data line by line and then outputs new key-value pairs: $<k_3, v_3>$.

2.2. Overview. The workflow of the proposed method is as Figure 1 shows. The input is the biomedical documents, and the output is the document cluster. The details are as follows:

(1) *Preprocessing*. The first step is to extract the semantic features of each document. The second step is to transform the data to put together documents that have the same semantic feature by using MapReduce

(2) *MapReduce-Based Semantic Similarity Calculation*. Calculate the MeSH heading similarity in advance and then calculate the document similarity with the average maximum match

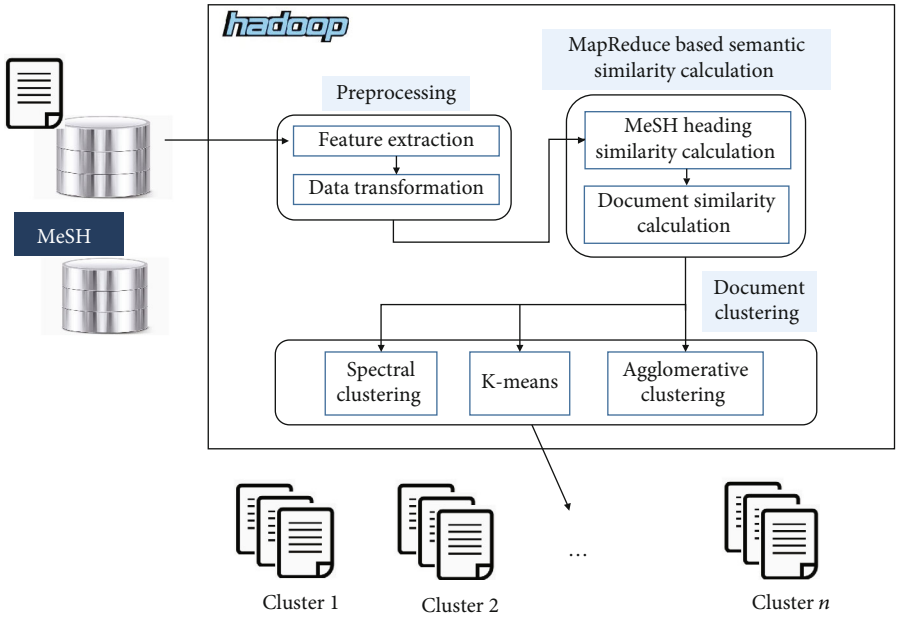

Figure 1: The workflow of the proposed method.

```
Data transformation
Input: <d, list(m)>
Output: <m, list(d)>
Notation: Write (k, v) outputs <k, v>
Class mapper
   Method map (d, list(m))
      For each m ∈ list(m)
         Write (m, d)
      End for
Class reducer
   Method reduce (m, list(d))
         s ← string(list(d))
         Write (m, s)
```

Algorithm 1: Algorithm of MapReduce-based data transformation.

```
MeSH heading similarity calculation
Input: <m, list(nodes)>
Output: <pair of m, semantic similarity>
Notation: Write (k, v) outputs <k, v>
Class mapper
   Method map (m, list(nodes))
      m1 ← MeSH heading
      For each m2 ∈ M
         r ← Sim (m1, m2)
         s ← string (m1 +" &" + m2)
         Write (s, r)
      End for
```

Algorithm 2: Algorithm of MapReduce-based MeSH heading similarity calculation.

(3) *Document Clustering.* Apply the cluster algorithm over the document similarity. In this article, we perform *K* -means, agglomerative clustering, and spectral clustering, respectively, over the document similarity

2.3. Preprocessing. In MEDLINE, each document is associated with a unique PubMed ID (PMID) and is tagged with several MeSH headings. Since the MeSH headings describe the subject of the documents, the MeSH headings can be

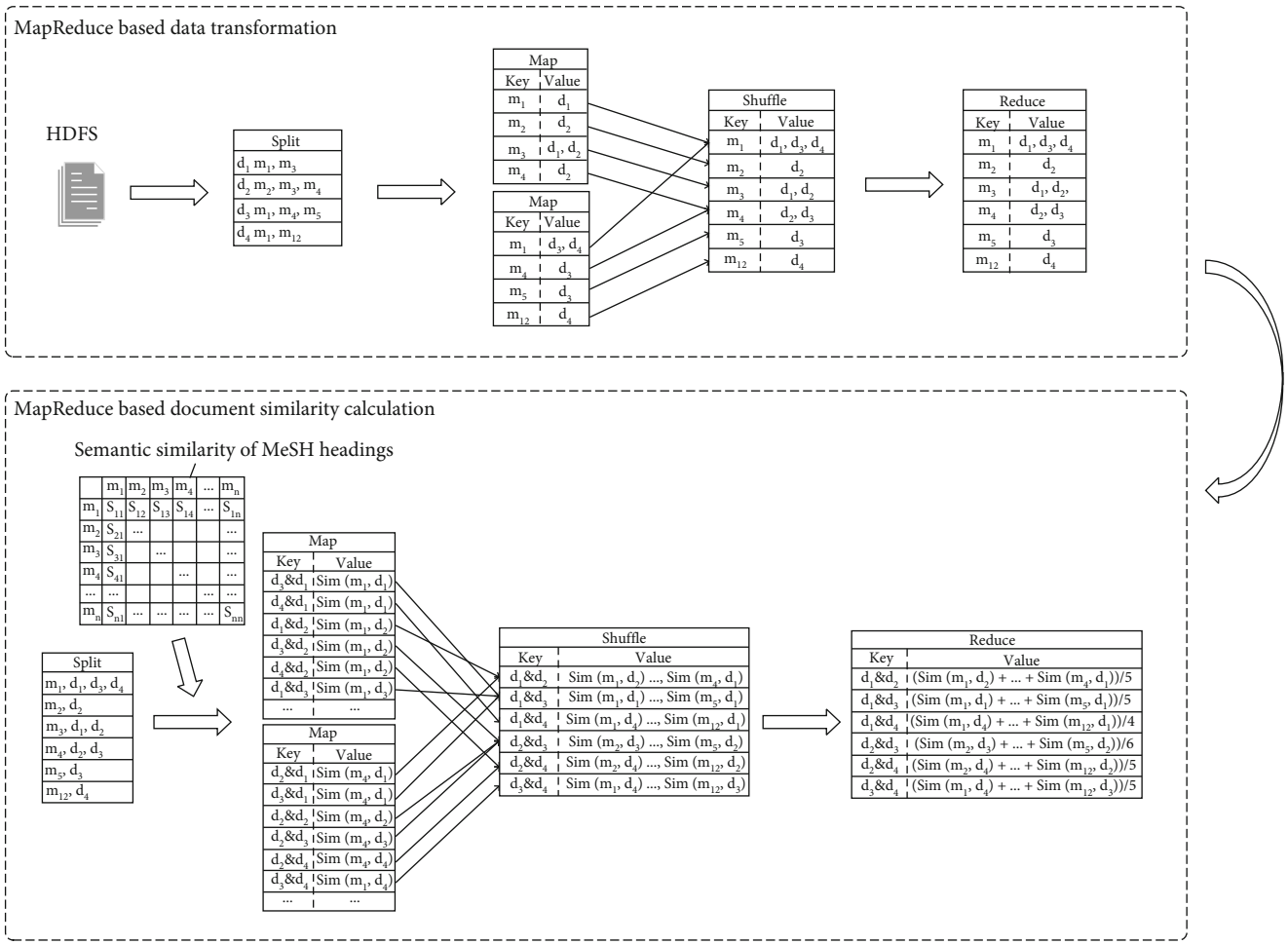

Figure 2: An example of MapReduce-based data transformation and semantic similarity calculation.

```
Document similarity calculation
Input: <m, list(d)>
Output: <pair of d, similarity>
Notation: Write (k, v) outputs <k, v>
Class mapper
   Method map (heading, list(d))
      m ← heading
      For each d1 ∈ D
         r ← Sim (m, d1)
         For each d2 in list(d)
            s ← string (d1 +" &" + d2)
            Write (s, r)
         End for
      End for
Class reducer
   Method reduce (s, list(r))
Sum ← 0, count ← 0
   For each r in list(r)
         Sum ← sum + r
         Count ← count +1
      End for
      Write (s, sum/count)
```

Algorithm 3: Algorithm of MapReduce-based document similarity calculation.

Table 2: Summary of the dataset SL.

	Documents	Classes	Unique MeSH headings	Total MeSH headings
Min	51	3	387	1619
Max	1619	12	2502	25631
Mean	689	7.5	1458	2502

Table 3: Summary of the dataset LUs.

	Documents	Unique MeSH headings	Total MeSH headings
Min	10000	14499	123347
Max	60000	25742	731089
Mean	35000	21540	427731

viewed as semantic features. Furthermore, the semantic similarity between documents can be represented by the semantic similarity between the sets of MeSH headings. Zhu et al. [37] and Zhou et al. [38] have proved the feasibility of this method. Therefore, we first extract the corresponding MeSH headings of documents through Efetch in NCBI. To put together documents that have the same semantic features, we transform the input document denoted by "$d_k \# m_{k_1}, m_{k_2}, m_{k_3}$" into the format "$m_1 \# d_1, d_2, d_4$" which means that the documents "d_1," "d_2," and "d_4" contain the same MeSH heading "m_1." The output is denoted as $<m$, list(PMID) $>$, where m is a MeSH heading.

The data transformation algorithm is as follows:

2.4. Semantic Similarity Calculation. To compute the semantic contribution of each MeSH heading to the document, we use Wang's average maximum match (AMM) strategy [39]. In Wang's study, Wang used AMM strategy to compute the semantic similarity between two sets of Gene Ontology (GO) terms. Since the AMM strategy is able to accurately detect the similarity between sets of semantic features, we applied AMM strategy to compute the semantic similarity between two sets of MeSH headings. The AMM strategy is defined as follows:

$$\mathrm{Sim}\left(d, d'\right) = \frac{\sum_{m \in d} \mathrm{Sim}\left(m, d'\right) + \sum_{m'^{\in d'}} \mathrm{Sim}\left(m', d\right)}{\mathrm{MeSHNumber}(d) + \mathrm{MeSHNumber}\left(d'\right)}, \quad (1)$$

$$\mathrm{Sim}(m, d) = \max_{m' \in d} \mathrm{Sim}\left(m, m'\right), \quad (2)$$

$$\mathrm{Sim}\left(m, m'\right) = \frac{\sum_{v \in m} \mathrm{Sim}\left(v, m'\right) + \sum_{v'^{\in m'}} \mathrm{Sim}\left(v', m\right)}{\mathrm{NodeNumber}(m) + \mathrm{NodeNumber}\left(m'\right)}, \quad (3)$$

$$\mathrm{Sim}(v, m) = \max_{v' \in m} \mathrm{Sim}^{\mathrm{sem}}\left(v, v'\right), \quad (4)$$

where NodeNumber() returns the node number of the MeSH heading, and MeSHNumber() returns the MeSH heading number of the document. The semantic measure is optional. The measures used in this paper are as follows:

(1) SP [27]

$$\mathrm{Sim}^{\mathrm{SP}}\left(m, m'\right) = \frac{L_{\max}\left(m, m'\right) - L_{\min}\left(m, m'\right)}{L_{\max}\left(m, m'\right)}, \quad (5)$$

where $L_{\max}(m, m')$ returns the longest path length, and $L_{\min}(m, m')$ returns the shortest path length.

(2) WP [28]

$$\mathrm{Sim}^{\mathrm{WP}}\left(v, v'\right) = \frac{2 * \mathrm{depth}\left(lca\left(v, v'\right)\right)}{\mathrm{depth}(v) + \mathrm{depth}\left(v'\right)}, \quad (6)$$

where depth(v) returns the tree depth of the node.

(3) LC [31]

$$\mathrm{Sim}^{\mathrm{LC}}\left(m, m'\right) = 1 - \frac{\log\left(1 + L_{\min}\left(m, m'\right)\right)}{\log\left(1 + 2D\right)}, \quad (7)$$

where D is the maximum depth of the heading in MeSH ontology.

(4) Res [29]

$$\mathrm{Sim}^{\mathrm{Res}}\left(v, v'\right) = \mathrm{IC}\left(lca\left(v, v'\right)\right). \quad (8)$$

(5) Lin [30]

$$\mathrm{Sim}^{\mathrm{Lin}}\left(v, v'\right) = \frac{2 * \mathrm{IC}\left(lca\left(v, v'\right)\right)}{\mathrm{IC}(v) + \mathrm{IC}\left(v'\right)}. \quad (9)$$

(6) Sch [40]

$$\mathrm{Sim}^{\mathrm{Sch}}\left(v, v'\right) = \frac{2 * \mathrm{IC}\left(lca\left(v, v'\right)\right)}{\mathrm{IC}(v) + \mathrm{IC}\left(v'\right)} * \left(1 - \exp\left(-\mathrm{IC}\left(lca\left(v, v'\right)\right)\right)\right), \quad (10)$$

$$\mathrm{IC}(v) = H(v) * \left(1 - \frac{\log\left(|C(v)| + 1\right)}{\log\left(T_{\mathrm{total}}\right)}\right). \quad (11)$$

IC(v) returns the IC value of the node. $H(v)$ returns the depth of the node in the ontology. $C(v)$ is the set of the children of the node, and T_{total} is the total node number of the ontology.

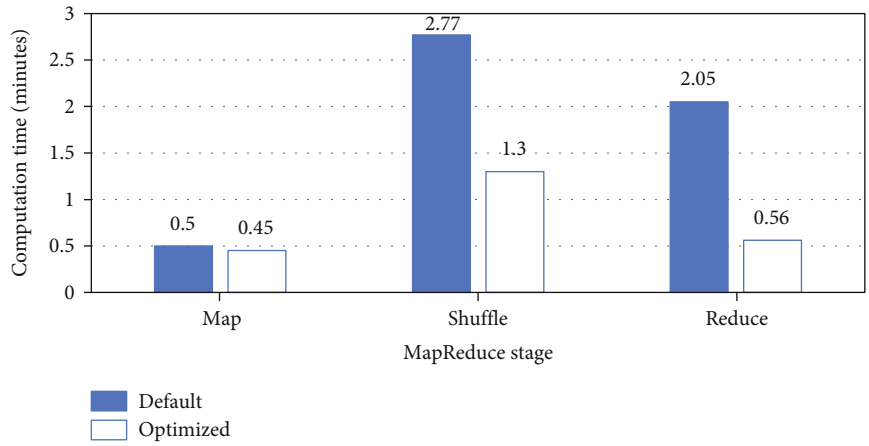

Figure 3: The result of MapReduce job optimization.

Table 4: Configuration of computers.

Configuration	Details
OS	Centos 7
CPU	I5-6500, 3.2 GHz
HDD	1 TB, 7200 rpm
RAM	8 G
Hadoop	Hadoop 3.1.3
JDK	1.8.0_252

We use SORA [41] to calculate the IC value. It is an ontology structure-based method, outperforming the corpus-based method on computation time.

According to AMM, semantic similarity calculation is divided into MeSH heading similarity calculation and document similarity calculation. Since the MeSH heading similarity is frequently used when computing the document similarity, we calculate the similarity of all pairs of extracted MeSH headings in advance. The MapReduce-based MeSH heading similarity calculation algorithm is as follows:

Traditionally, document similarity is calculated pair by pair, leading to large computational cost, which is the main reason why the existing methods can hardly work with a large number of documents. To make the AMM applied to the parallelization condition, we designed a MapReduce-based algorithm to calculate the document similarity in parallel. In this method, the similarity contribution of a MeSH heading to a document is viewed as a basic computation element. By splitting the semantic similarity between documents into the aggregation of multiple heading-to-document similarity denoted as $\mathrm{sim}(m, d)$, we realized the parallel computation of the document similarity. In addition, for each line of input, we directly output the semantic similarity of the MeSH heading to other documents, avoiding redundant computation. The algorithm is as follows, and an example is given in Figure 2:

Supposing that the number of documents is n, average MeSH headings number of documents is m, the total number of MeSH headings is k, and the time complexity of the traditional method is $O(m^2n^2)$. For the proposed MapReduce-based algorithm, the time complexity of map stage is $O(kmn)$, and the time complexity of reduce stage is $O(n^2)$.

2.5. Document Clustering. Spectral clustering [42, 43], agglomerative clustering [44, 45], and K-means [46, 47] are commonly used in text clustering. Spectral clustering is based on graph, which transforms the clustering problem into the optimal partition problem of a graph by treating each document as the vertex of the graph and the similarity between documents as the edge weight. The clusters are obtained by cutting the graph according to some rules such as Ncut [48] and Mcut [49]. Agglomerative clustering treats each document as a cluster first and then merges the most similar cluster repeatedly. K-means is carried out through multiple iterations. In each iteration, each document is divided into the most similar cluster until the cluster no longer changes. In this paper, these three clustering algorithms are performed, respectively, over the document similarity.

2.6. Data. For the analysis of multiangle, two kinds of datasets were applied in this experiment. One is a small and labelled dataset named SL used for verifying the accuracy of the proposed method. The other one is large and unlabelled dataset named LUs, being used for testing the efficiency of the method when dealing with a large number of documents.

SL is generated from Text REtrieval Conference (TREC) genomics track 2005, which contains biomedical documents with 50 topics. In TREC genomics track, each document is judged as definitely relevant (DR), possibly relevant (PR), or not relevant (NR) to the topic. We remove the PR and NR documents, reserving DR documents.

When generating the data set, we referred to the practice of Gu et al. [50]. To avoid small clusters, we remove the topics that have less than 10 documents. Furthermore, we remove the documents that are relevant to 2 or more topics. Finally, the dataset of 2,317 documents with 24 topics were obtained. We randomly selected documents of 3-12 topics to generate 100 different datasets. Table 2 shows the summary of these datasets.

Table 5: Computation time (minutes) of the traditional and proposed method with different semantic measures ("/" means cannot work).

Method	Dataset: SL						Dataset: LUs-10000					
	SP	WP	LC	Res	Lin	Sch	SP	WP	LC	Res	Lin	Sch
Traditional	68.9	66.35	67.9	61.15	64.8	66.25	/	/	/	/	/	/
Proposed	2.87	1.02	1.42	1.15	0.95	1.17	10.21	3.43	5.02	2.31	3.30	3.77

Table 6: NMI (Average ± Standard Deviation) on dataset SL with different cluster algorithms and semantic measures.

Cluster algorithm	SP	WP	LC	Resnik	Lin	Sch
Spectral clustering	0.579 ± 0.126	0.549 ± 0.132	0.574 ± 0.130	0.647 ± 0.124	0.617 ± 0.111	0.527 ± 0.124
K-means	0.490 ± 0.140	0.495 ± 0.131	0.491 ± 0.134	0.511 ± 0.176	0.526 ± 0.158	0.520 ± 0.123
Agglomerative clustering	0.524 ± 0.123	0.551 ± 0.145	0.533 ± 0.124	0.591 ± 0.116	0.582 ± 0.135	0.523 ± 0.126
Zhu	/	0.568 ± 165	0.565 ± 0.169	/	0.620 ± 0.161	/

LUs include six datasets randomly extracted 10000 to 60000 documents from PubMed, covering more than 20000 different MeSH headings. For each dataset, we mark it with the number of documents, such as LUs-10000 and LUs-20000. Table 3 is the summary of the dataset LUs.

2.7. Evaluation Criteria. In the experiment, the performance is evaluated by comparing the predicted label and the true label. We take Normalized Mutual Information (NMI) as the evaluation index, since it has been proved that NMI outperforms many other clustering evaluation indexes [40]. The NMI formula [41] is defined as follows:

$$\mathrm{NMI} = \frac{\sum_{h,l} n_{h,l} \log\left(n \cdot n_{h,l} / n_h n_l\right)}{\sqrt{\left(\sum_h n_h \log\left(n_h/n\right)\right)\left(\sum_l n_l \log\left(n_l/n\right)\right)}}, \quad (12)$$

where n is the total number of documents to be clustered, n_h is the number of documents with true class h, n_l is the number of documents with predicted class l, and $n_{h,l}$ is the number of documents with true class h and predicted class l.

NMI ranges from 0 to 1. A high NMI value means the strong correlation between the predicted label and the true label.

2.8. Experimental Environment. The hardware and software details of each computer are shown in the following table.

The experimental environment is a Hadoop cluster composed of five computers with the same configuration.

3. Results and Discussion

3.1. Optimization of MapReduce Job Settings. Before testing the efficiency of the proposed method with a large number of documents, MapReduce job settings are optimized according to the characteristics of the proposed method since the job settings have a great impact on task execution [51]. It can be easily found that the input and output of the proposed method are very compact, while a lot of intermediate key-value pairs are generated in the Map task during MapReduce-based semantic similarity calculation, which will generate much data to be sorted and aggregated in Shuffle. Therefore, according to the MapReduce optimization principle of multiset homomorphisms proposed by Dorre et al. [51], we increase the number of reduce tasks to enhance the parallelism and add the combiner used to aggregate the data before shuffle.

As is shown in Figure 3, on the dataset LUs-10000 with Resnik measure, the elapsed time of map is reduced from 0.5 minutes to 0.45 minutes, the elapsed time of shuffle is reduced from 2.77 minutes to 1.3 minutes, and the elapsed time of reduce is reduced from 2.05 minutes to 0.56 minutes. The result shows that the optimization reduces the intermediate data to be shuffled and promotes the efficiency of reduce, effectively decreasing the computation time. And we used the same optimized job settings in the following experiments.

3.2. Evaluation of Clustering and Computation Efficiency. In the experiment, the proposed method was conducted with six semantic measures and three cluster algorithms. Then, we compared the traditional method with the proposed method on both computation time and NMI. Tables 5 and 6 show the results.

(A) Computational efficiency

For the small dataset SL, the traditional method takes more than an hour, while the proposed method takes no more than 3 minutes with the cluster of five computers. For the big dataset LUs-10000, the traditional method can hardly work due to the big data, while the proposed method keeps efficient. Various semantic measures are available in this method, and the IC-based methods take less time than other methods.

(B) Clustering validation

For the spectral clustering, the highest NMI of 0.647 is achieved with Resnik. For the K-means, the highest NMI of 0.526 is achieved with Lin. For the agglomerative clustering, the highest NMI of 0.591 is achieved with Resnik. The result reveals that the information content-based measure (Resnik and Lin) outperforms other semantic measures, and spectral cluster performs better than the other two cluster algorithms. The highest NMI is obtained by Resnik measure and spectral clustering algorithm. Compared with the

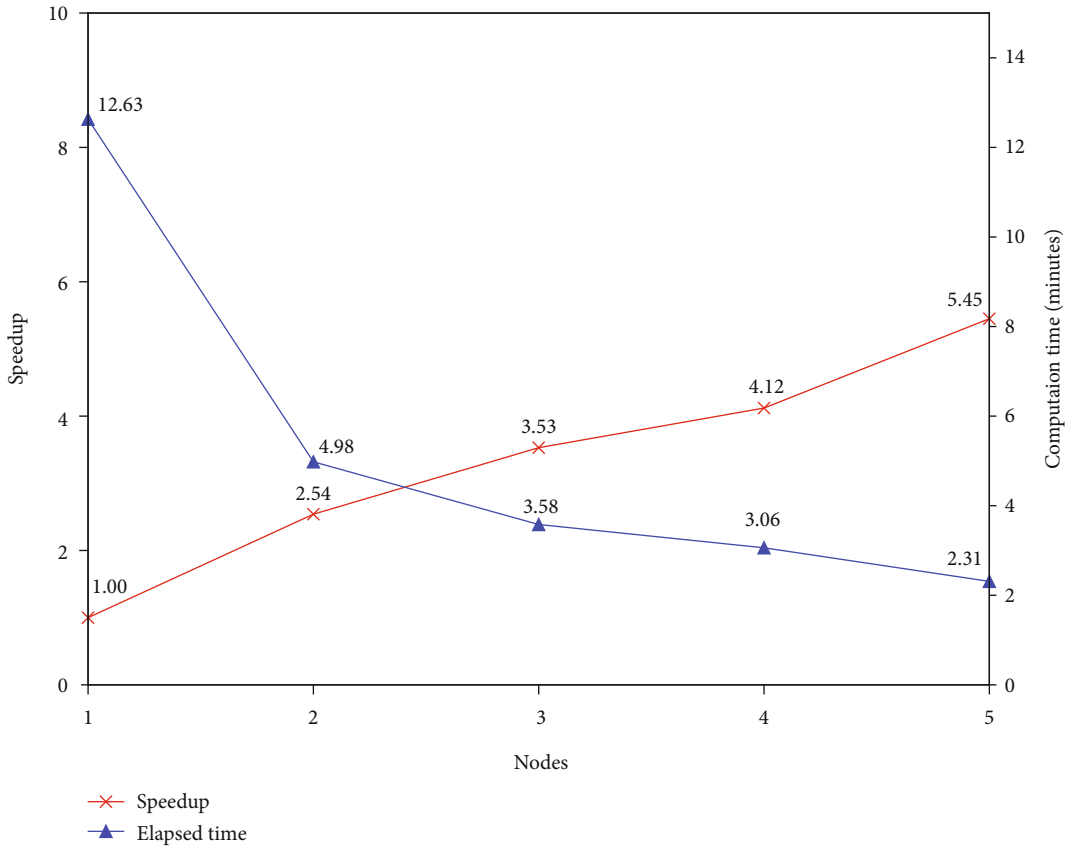

Figure 4: The trend of speedup and computation time with increasing cluster nodes.

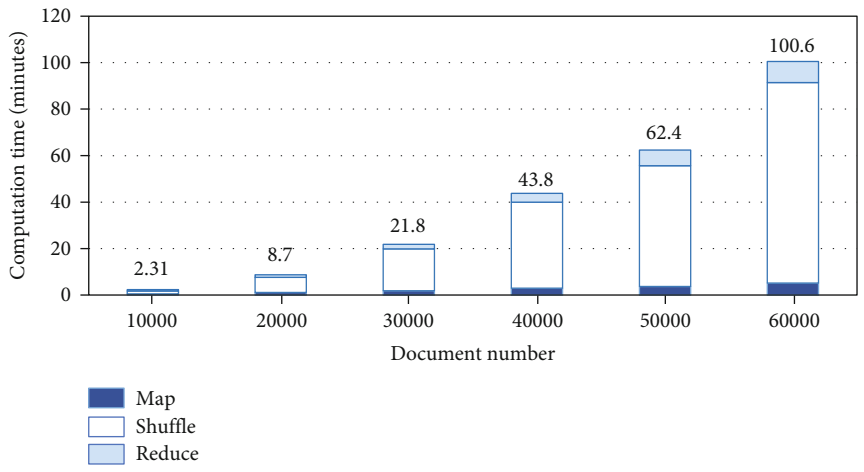

Figure 5: Computation time of map, shuffle, and reduce on dataset LUs.

result in Zhu et al.'s study [37] where the same data and evaluation criteria were used, NMI of the proposed method is slightly increased, implying that the proposed method greatly improves the computational efficiency without decreasing the clustering accuracy.

3.3. Speedup and Elapsed Time with Different Cluster Node Number. To study the parallelism of the method, the proposed method was performed with different cluster node number on dataset LUs of 10000 documents. Figure 4 shows that the elapsed time goes down from 12.63 minutes to 2.31 minutes, and the speedup goes up almost linearly from 1 to 5.45 as the cluster nodes increase, implying that the proposed method is of high parallelism, and increasing nodes will improve the computation time effectively.

3.4. Computation Time with More Documents. In this section, the experiment was performed on the dataset LUs to

observe the trend of the elapsed time and the proportion of each stage in the MapReduce job. Figure 5 shows that the proposed method remains effective when processing a big number of documents. As the documents increase, the elapsed time of map and reduce grows slowly while the elapsed time consumed in shuffle grows rapidly. And shuffle accounts for the largest proportion of computation time in all MapReduce tasks. The result reveals that the sort and copy of data become the key factor to the computation time when processing a big number of documents.

4. Conclusions

In this paper, we developed an efficient ontology-based semantic similarity measure for big document data clustering. Traditionally, the semantic similarity between documents is computed in pairwise, which can hardly work with a big number of documents. To solve the problem, we developed a MapReduce-based method to process the data in parallel. By splitting the document similarity into the aggregation of multiple heading-to-document similarity, the proposed method avoids the redundant computation and is available to process a big number of documents in a short time. Additionally, according to the experiment results, it can be concluded that the proposed method is of high parallelism and scalability, implying that more documents can be processed as long as we increase the cluster nodes, upgrade the hardware of computers, and optimize the job settings properly. In this work, both semantic measure and the cluster algorithm are optional, which depend on the datasets and the ontology. For the TREC 2005 genomics track dataset and MeSH ontology, the spectral algorithm and the semantic measure of Resnik perform better than other parameters. Furthermore, the proposed method is not limited to biomedical documents and MeSH ontology. The proposed method can also work in the situation of combining semantic similarity from different semantic features.

Data Availability

The data are available from Text REtrieval Conference (TREC, https://trec.nist.gov/) and PubMed (https://pubmed.ncbi.nlm.nih.gov/).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61911540482 and 61702324).

References

[1] V. H. A. Soares, R. J. G. B. Campello, S. Nourashrafeddin, E. Milios, and M. C. Naldi, "Combining semantic and term frequency similarities for text clustering," *Knowledge and Information Systems*, vol. 61, no. 3, pp. 1485–1516, 2019.

[2] M. S. Tajbakhsh and J. Bagherzadeh, "Semantic knowledge LDA with topic vector for recommending hashtags: Twitter use case," *Intelligent Data Analysis*, vol. 23, no. 3, pp. 609–622, 2019.

[3] H. T. Wang, K. K. Tian, Z. J. Wu, and L. Wang, "A short text classification method based on convolutional neural network and semantic extension," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 367–375, 2021.

[4] N. M. Ranjan and R. S. Prasad, "Automatic text classification using BPLion-neural network and semantic word processing," *Imaging Science Journal*, vol. 66, no. 2, pp. 69–83, 2018.

[5] B. Sahoh and A. Choksuriwong, "Automatic semantic description extraction from social big data for emergency management," *Journal of Systems Science and Systems Engineering*, vol. 29, no. 4, pp. 412–428, 2020.

[6] X. Cen, J. Yuan, C. Pan, Q. Tang, and Q. Ma, "Contextual embedding bootstrapped neural network for medical information extraction of coronary artery disease records," *Medical & Biological Engineering & Computing*, vol. 59, no. 5, pp. 1111–1121, 2021.

[7] Q. Qiu, Z. Xie, L. Wu, and L. Tao, "Automatic spatiotemporal and semantic information extraction from unstructured geoscience reports using text mining techniques," *Earth Science Informatics*, vol. 13, no. 4, pp. 1393–1410, 2020.

[8] N. Gao, Z. Y. Zhu, Z. Q. Weng, G. L. Chen, and M. Zhang, "A supervised named entity recognition method based on pattern matching and semantic verification," *Journal of Internet Technology*, vol. 21, no. 7, pp. 1917–1928, 2020.

[9] C. Y. Wang, H. Wang, H. Zhuang et al., "Chinese medical named entity recognition based on multi-granularity semantic dictionary and multimodal tree," *Journal of Biomedical Informatics*, vol. 111, p. 103583, 2020.

[10] B. Ji, S. Li, J. Yu, J. Ma, and Y. Ji, "Research on Chinese medical named entity recognition based on collaborative cooperation of multiple neural network models," *Journal of Biomedical Informatics*, vol. 104, no. S3, article 103395, 2020.

[11] K. Chakraborty, S. Bhatia, S. Bhattacharyya, J. Platos, R. Bag, and A. E. Hassanien, "Sentiment analysis of COVID-19 tweets by deep learning classifiers–a study to show how popularity is affecting accuracy in social media," *Applied Soft Computing*, vol. 97, no. Part A, article 106754, 2020.

[12] M. G. Huddar, S. S. Sannakki, and V. S. Rajpurohit, "Multi level feature optimization and multimodal contextual fusion for sentiment analysis and emotion classification," *Computational Intelligence*, vol. 36, no. 2, pp. 861–881, 2020.

[13] H. Zhang, J. Dong, L. Min, and P. Bi, "A BERT fine-tuning model for targeted sentiment analysis of Chinese online course reviews," *International Journal on Artificial Intelligence Tools*, vol. 29, no. 7n08, article 2040018, 2020.

[14] A. Sahu and P. K. Bhowmick, "Feature engineering and ensemble-based approach for improving automatic short-answer grading performance," *IEEE Transactions on Learning Technologies*, vol. 99, p. 1, 2019.

[15] W. Davy, S. Abeed, K. Ari, O. Karen, M. Arjun, and G. H. Graciela, "Deep neural networks ensemble for detecting medication mentions in tweets," *Journal of the American Medical Informatics Association*, vol. 26, no. 12, pp. 1618–1626, 2019.

[16] J. Li, X. Zhang, and X. Zhou, "ALBERT-based self-ensemble model with semisupervised learning and data augmentation

for clinical semantic textual similarity calculation: algorithm validation study," *JMIR Medical Informatics*, vol. 9, no. 1, article e23086, 2021.

[17] X. Yang, X. He, H. Zhang, Y. Ma, J. Bian, and Y. Wu, "Measurement of semantic textual similarity in clinical texts: comparison of transformer-based models," *JMIR Medical Informatics*, vol. 8, no. 11, article e19735, 2020.

[18] S. M. Subramanian and V. Soundarajan, "SC-OCR: similarity-based clustering and optimum cache replacement approach," *Concurrency and Computation-Practice & Experience*, vol. 29, no. 4, 2017.

[19] R. Vinaitheerthan, "Text mining in biomedical domain with emphasis on document clustering," *Healthcare Informatics Research*, vol. 23, no. 3, pp. 141–146, 2017.

[20] A. Onan, H. Bulut, and S. Korukoglu, "An improved ant algorithm with LDA-based representation for text document clustering," *Journal of Information Science*, vol. 43, no. 2, pp. 275–292, 2017.

[21] J. Peng, D. Q. Yang, S. W. Tang, T. J. Wang, and J. Gao, "A new similarity computing method based on concept similarity in Chinese text processing," *Science in China Series F-Information Sciences*, vol. 51, no. 9, pp. 1215–1230, 2008.

[22] C. S. Tasi, Y. M. Huang, C. H. Liu, and Y. M. Huang, "Applying VSM and LCS to develop an integrated text retrieval mechanism," *Expert Systems with Applications*, vol. 39, no. 4, pp. 3974–3982, 2012.

[23] J. Liu, L. Lin, H. L. Ren et al., "Building neural network language model with POS-based negative sampling and stochastic conjugate gradient descent," *Soft Computing*, vol. 22, no. 20, pp. 6705–6717, 2018.

[24] J. Flisar and V. Podgorelec, "Improving short text classification using information from DBpedia ontology," *Fundamenta Informaticae*, vol. 172, no. 3, pp. 261–297, 2020.

[25] A. Khan, N. Salim, H. Farman et al., "Abstractive text summarization based on improved semantic graph approach," *International Journal of Parallel Programming*, vol. 46, no. 5, pp. 992–1016, 2018.

[26] B. Sathiya and T. V. Geetha, "A review on semantic similarity measures for ontology," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 4, pp. 3045–3059, 2019.

[27] H. Bulskov, R. Knappe, and T. Andreasen, "On measuring similarity for conceptual querying," *Flexible Query Answering Systems, Proceedings*, vol. 2522, pp. 100–111, 2002.

[28] Y. Y. Wang, "Verb semantics and lexical selection," *Computer Science*, vol. 14, no. 101, pp. 325–327, 1994.

[29] P. Resnik, "Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research*, vol. 11, pp. 95–130, 1999.

[30] D. Lin, *An Information-Theoretic Definition of Similarity*, 1998.

[31] C. Leacock and M. Chodorow, *Combining Local Context and WordNet Similarity for Word Sense Identification*, MIT Press, 1998.

[32] Y. H. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871–882, 2003.

[33] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *Rocling*, no. article 11512, 1997.

[34] C. G. Zhao and Z. Wang, "GOGO: an improved algorithm to measure the semantic similarity between gene ontology terms," *Scientific Reports*, vol. 8, 2018.

[35] Y. J. Zhang, Q. Y. Chen, Z. H. Yang, H. F. Lin, and Z. Y. Lu, "BioWordVec, improving biomedical word embeddings with subword information and MeSH," *Scientific Data*, vol. 6, no. 1, p. 52, 2019.

[36] A. Rui and M. Sérgio, "Supervised learning and knowledge-based approaches applied to biomedical word sense disambiguation," *Journal of Integrative Bioinformatics*, vol. 14, no. 4, 2017.

[37] S. F. Zhu, J. Zeng, and H. Mamitsuka, "Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity," *Bioinformatics*, vol. 25, no. 15, pp. 1944–1951, 2009.

[38] J. Zhou, Y. X. Shui, S. W. Peng, X. H. Li, H. Mamitsuka, and S. F. Zhu, "MeSHSim: an R/Bioconductor package for measuring semantic similarity over MeSH headings and MEDLINE documents," *Journal of Bioinformatics and Computational Biology*, vol. 13, no. 6, 2015.

[39] J. Z. Wang, Z. D. Du, R. Payattakool, P. S. Yu, and C. F. Chen, "A new method to measure the semantic similarity of GO terms," *Bioinformatics*, vol. 23, no. 10, pp. 1274–1281, 2007.

[40] A. Schlicker, F. S. Domingues, J. Rahnenfuhrer, and T. Lengauer, "A new measure for functional similarity of gene products based on Gene Ontology," *Bmc Bioinformatics*, vol. 7, 2006.

[41] Z. X. Teng, M. Z. Guo, X. Y. Liu, Q. G. Dai, C. Y. Wang, and P. Xuan, "Measuring gene functional similarity based on group-wise comparison of GO terms," *Bioinformatics*, vol. 29, no. 11, pp. 1424–1432, 2013.

[42] R. Janani and S. Vijayarani, "Text document clustering using spectral clustering algorithm with particle swarm optimization," *Expert Systems with Applications*, vol. 134, pp. 192–200, 2019.

[43] N. Passalis and A. Tefas, "Information clustering using manifold-based optimization of the bag-of-features representation," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 52–63, 2018.

[44] A. Naeem, M. Rehman, M. Anjum, and M. Asif, "Development of an efficient hierarchical clustering analysis using an agglomerative clustering algorithm," *Current Science*, vol. 117, no. 6, pp. 1045–1053, 2019.

[45] T. Jo, "Clustering texts using feature similarity based AHC algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 6, pp. 5993–6003, 2018.

[46] S. Kongwudhikunakorn and K. Waiyamai, "Combining distributed word representation and document distance for short text document clustering," *Journal of Information Processing Systems*, vol. 16, no. 2, p. 277, 2020.

[47] F. Yang, G. S. Liu, K. Meng, and Z. Y. Sun, "Neural feedback text clustering with BiLSTM-CNM-Kmeans," *IEEE Access*, vol. 6, pp. 57460–57469, 2018.

[48] J. B. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[49] M. Meila and J. Shi, "A random walks view of spectral segmentation," *8th International Workshop on Artificial Intellignece and Statistics*, 2001.

[50] J. Gu, W. Feng, J. Zeng, H. Mamitsuka, and S. F. Zhu, "Efficient semisupervised MEDLINE document clustering with MeSH-semantic and global-content constraints," *IEEE Transactions on Cybernetics*, vol. 43, no. 4, pp. 1265–1276, 2013.

[51] J. Dorre, S. Apel, and C. Lengauer, "Modeling and optimizing MapReduce programs," *Concurrency and Computation-Practice & Experience*, vol. 27, no. 7, pp. 1734–1766, 2015.