International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307

Semantic similarity based clustering and modeling using Latent Dirichlet Allocation (LDA)

Anusha N, Soumya Bilagi, Raghava M S
Department of Computer Science and Engineering, Sambhram Institute Of Technology, M S Palya, Banglore, Karnataka, India

ABSTRACT

Privacy has become a substantial issue once the applications of big data are dramatically growing in cloud computing. In recent years, we have a tendency to focus on privacy and propose a unique a novel approach that is termed Dynamic data encryption Strategy (D2ES). Our planned approach aims to by selection encipher knowledge and use privacy classification ways under temporal order constraints. This approach is intended to maximize the privacy protection scope by employing a selective coding strategy within the specified execution time necessities. During this paper, is intended victimization semantic similarity based mostly clustering and topic modeling victimization Latent Dirichlet Allocation (LDA) for summarizing the big text collection over Map reduce framework. The account task is performed in four stages and provides a standard implementation of multiple documents account. The conferred technique is evaluated in terms of quantifiability and varied text account parameters particularly, compression ratio, retention ratio, ROUGE and Pyramid score are measured. The benefits of Map scale back framework are clearly visible from the experiments and it's additionally incontestable that Map reduce provides a quicker implementation of summarizing giant text collections and may be a powerful tool in big Text data analysis.

Keywords: Map Reduce, Latent Dirichlet Allocation (LDA), encryption, clustering

I. INTRODUCTION

Text summarization is one in all the vital and difficult issues in text mining. It provides variety of advantages to users and variety of fruitful real world applications is developed using text summarization. In text summarization an outsized collections of text documents area unit reworked to a reduced and compact text document that represents the digest of the first text collections. A summarized document helps in understanding the gist of big text collections quickly and conjointly save tons of your time by avoiding reading of every individual document in a very large text collection.

Multi-document summarization may be a technique wont to summarize multiple text documents and is employed for understanding massive text document collections. Multi-document summarization generates a compact outline by extracting the relevant sentences from a group of documents on the premise of document topics. Within the recent years researchers have given abundant attention towards developing document summarization techniques. Variety of summarization techniques area unit projected to come up with summaries by extracting the vital sentences from the given collection of documents. Multi-document summarization is employed for understanding and analysis of huge document collections, the most important supply of those collections area unit news archives, blogs, tweets, web pages, analysis papers, internet search results and technical reports on the market over the net and different places. Some samples of the applications of the Multi-document summarization are analyzing the net search results for helping users in any browsing, and generating summaries for news articles. Document process and outline generation in a very massive text document assortment is computationally complicated task and within the era of huge knowledge analytics wherever size of knowledge collections is high there's would like of algorithms for summarizing the big text collections quickly. During this paper, a Map Reduce framework based mostly summarization methodology is projected to come up with the summaries from massive text collections. Experimental results on UCI machine learning repository knowledge sets reveal that the procedure time for summarizing massive text collections is drastically reduced using the Map Reduce framework and Map Reduce provides measurability for accommodating massive text collections for summarizing. Performance measuring metric of summarization ROUGE and Pyramid scores are provides acceptable values in summarizing the big text collections.

Single-document summarization is easy to handle since just one text document must be analyzed for summarization, whereas handling multi-document summarization could be an advanced and tough task. It needs variety of (multiple) text documents to be analyzed for generating a compact and informative (meaningful) outline. Because the variety of documents will increase in multi-document account, the summarizer gets additional difficulties in playing the account. A summarizer is alleged to be smart, if it contains additional fruitful and relevant compact representation of large text collections. Considering semantic similar terms offer edges in terms of generating additional relevant outline however it's additional reason intensive, since semantic terms are generated and thought of for making outline from an oversized text assortment. During this work the issues with multi document text account are self-addressed with the assistance of latest technologies in text analytics. A multi-document summarizer is given during this work with the assistance of semantic similarity primarily based cluster over the popular distributed computing framework Map Reduce.

II. ALGORITHM

K-means clustering algorithm

Clustering is a process of creating groups of similar objects. Clustering algorithms are categorized into five major categories namely, Partitioning techniques, Hierarchical techniques, Density Based techniques, Grid Based techniques and Model based techniques. Partitioning techniques are the simplest techniques which creates K number of disjoint partitions to create K number of clusters. These partitions are created using certain statistical measures like mean, median etc. K-means is a classical unsupervised learning algorithms used for clustering. It is a simple, low complexity and a very popular clustering algorithm. The k-means algorithm is a partitioning based clustering algorithm. It takes an input parameter, k i.e. the number of clusters to be formed, which partitions a set of n objects to generate the k clusters. The algorithm works in three steps. In the first step, k number of the objects is selected randomly, each of which represents the initial mean or center of the cluster. In the second step, the remaining objects are assigned to the cluster with minimum distance from cluster center or mean. In the third step, the new mean for each cluster is computed and the process iterates until the criterion function converges.

The algorithm for proposed multi document summarization using semantic similarity based clustering technique is presented in this section. The algorithm is logically divided in four major stages; the algorithm for each stage is explained in this section. In the first stage of document summarization, the document clustering is performed using K-means clustering algorithm on Map Reduce framework. Mapper is responsible for part of documents and part of k centers. For each document, it finds closest of known centers and produces the output key as point, value identifies center and distance. Reducer takes minimum distance center and produces output key identifies center, value is document. A successive phase averages points in each center.

After creating the text document clustering, the document belonging to clusters are retrieved and text information present is each document is collected in aggregate. The topic modeling technique is then applied on collective information to generate the topics from each text document clusters. LDA (Latent Dirichlet Allocation) technique is used in this work for generating topics from each document cluster. In the third stage, semantic similar terms are computed for each topic term generated in previous stage. Word Net Java API is used to generate the list of semantic similar terms. The semantic similar terms are generated over the Map Reduce framework and the generated semantic terms are added to the vector. Semantic similar term finding is an intensive computing operation. It requires going through with the vocabulary and synonyms data for the given term in the hierarchy of semantic relationship. Map Reduce framework is utilized efficiently for handling this operation. The Mapper computes the semantic similar terms for each topic term generated by the document cluster and reducer aggregate these terms and counts the frequencies of these terms (topic terms and semantic similar terms of topic terms) aggregately.

Then the terms are arranged in the descending order of frequency and top N topic terms (including the semantic similar terms) are selected. These filtered terms are called as semantic similar frequent terms available in the document collection using the method Compute Semantic Similar(Ti) . The algorithm counts the number of occurrences of every word in a text collection. Input key-values pairs take the form of (document id, doc) pairs stored on the distributed file system. The key parameter is a unique identifier for the document, and the value parameter is the text of the document itself. The Mapper takes key-value pair as input, generates tokens from the document, and emits an intermediate key-value pair for every word. The Map Reduce execution makes sure that all values associated with the same key are brought together in the reducer. The final output of the algorithm is written to the distributed file system, one file per reducer. In the last stage, the original text document collection is distributed over the Mappers and using parsing techniques, sentences are extracted from individual document by the Mappers. The sentences which are consisting of the frequent terms and its semantic similar terms are filtered from the original text collection and added to the summary document (in other words the filtered terms participates in the summary document). The final summary is generated after traversing all the documents in the document collections.

III. CONCLUSION

A multi-document text summarizer supported Map Reduce framework is given in this work. Experiments are carried consumption to four nodes in Map Reduce framework for an oversized text collection and also the summarization performance parameters compression ratio, retention ratio and computation timings are evaluated for an oversized text collection. It is also shown experimentally that Map Reduce framework provides higher quantifiability and reduced time complexness whereas considering sizable amount of text documents for summarization. 3 possible cases of summarizing the multiple documents also are studied relatively. It's shown that effective report is performed once each clustering and linguistics similarity is thought-about. Considering semantic similarity provides better retention ratio, ROUGE and pyramid scores for summary.

IV. REFERENCES

[1]. Guoping Wang and CheeYong Chan, "MultiQuery Optimization in MapReduce Framework", 40th International Conference on Very Large Data Bases, September 2014.

[2]. Thomas Wirtz and Rong Ge, "Improving MapReduce Energy Efficiency for Computation Intensive Workloads", Green Computing Conference and Workshops (IGCC), 2011 International.

[3]. Prajesh P Anchalia, Anjan K Koundinya, Srinath N K, "MapReduce Design of K-Means Clustering Algorithm", 2013 IEEE.

[4]. Cheng T. Chu, Sang K. Kim, Yi A. Lin, Yuanyuan Yu, Gary R. Bradski, Andrew Y. Ng, and Kunle Olukotun, "Map-Reduce for Machine Learning on Multicore", NIPS, page 281--288. MIT Press, 2006.

[5]. Yaobin He, Haoyu Tan, Wuman Luo, Huajian Mao, Di Ma, Shengzhong Feng, Jianping Fan, "MR-DBSCAN: An Efficient Parallel Density-based Clustering Algorithm using MapReduce", 2011 IEEE 17th International Conference on Parallel and Distributed Systems.

[6]. J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan, "Web usage mining: discovery and applications of usage patterns from Web data," ACM SIGKDD Explorations Newsletter, vol. 1, 2000, pp. 12-23.

[7]. K. E. Amin and N. rouhani, "Web usage Mining:Discovery of the user's navigational patterns using SOM," IEEE, 2009.

[8]. N. Khasawneh and H. C. Chan, "Active User-Based and Ontology-Based Weblog data preprocessing for Web Usage Mining," presented at IEEE/WIC/ ACM International Conference, 2006.

[9]. S. A. Rios and J. D. Velasquez, "Semantic Web Usage Mining by a Concept-based aproach for Off-line Web Site Enhancements," presented at IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008.

[10]. M. A. Bayir, I. H. Toroslu, G. Fidan, and A. Cosar, "Smart Miner: A New Framework for Mining Large Scale Web Usage Data," ACM, 2009.

[11]. Shim K (2012) MapReduce Algorithms for Big Data Analysis, Framework. Proceedings of the VLDB Endowment 5(12):2016–2017

[12]. Lee K-H, Lee Y-J, Choi H, Chung YD, Moon B (2011) Parallel Data Processing with MapReduce: A Survey. ACM SIGMOD Record 40(4):11–20

[13]. Yang J, Li X (2013) MapReduce Based Method for Big Data Semantic Clustering. In Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference. Manchester, England, pp 2814–2819

[14]. Ene A, Im S, Moseley B (2011) Fast Clustering using MapReduce. Proc. of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, USA, pp 681–689

[15]. Kolb L, Thor A, Rahm E (2013) Don't Match Twice: Redundancy-free Similarity Computation with MapReduce. Proc. of the Second Workshop on Data Analytics in the Cloud, ACM, New York, USA, pp 1–5

[16]. Esteves RM, Rong C (2011) Using Mahout for clustering Wikipedia's latest articles: a comparison between K-means and fuzzy C-means in the cloud. In Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference. Athens, Greece, pp 565–569

[17]. Li HG, Wu GQ, Hu XG, Zhang J, Li L, Wu X (2011) K-means clustering with bagging and mapreduce. Proc. 2011 44th Hawaii International Conference on IEEE System Sciences (HICSS). Kauai/Hawaii, US, pp 1–8

[18]. Zhang G, Zhang M (2013) The Algorithm of Data Preprocessing in Web Log Mining Based on Cloud Computing. In 2012 International Conference on Information Technology and Management Science (ICITMS 2012) Proceedings Springer. Berlin, Heidelberg, Germany, pp 467–474

[19]. Morales GDF, Gionis A, Sozio M (2011) Social content matching in mapreduce. Proceedings of the VLDB Endowment 4(7):460–469

[20]. Verma A, Llora X, Goldberg DE, Campbell RH (2009) Scaling Genetic algorithms using MapReduce. Intelligent Systems Design and Application(ISDA). Ninth International Conference, Pisa, Italy, pp 13–18