



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

Truth Discovery in Social Media Applications Using Hadoop

Dhivya S¹, Gomathi M², Sakthi Sridevi J³, Mr Sathiya Jeba Sundar J S⁴

UG Scholar, Department of Computer Science and Engineering, Velammal Institute of Technology, Tiruvallur, Tamil Nadu, India^{1,2,3}

Assistant Professor, Department of Computer Science and Engineering, Velammal Institute of Technology, Tiruvallur, Tamil Nadu, India⁴

ABSTRACT- Perceiving dependable information inside seeing boisterous data contributed by different unvetted sources from online electronic life (e.g., Twitter, Facebook, and Instagram) has been an essential task in the season of gigantic data. This endeavor, suggested as truth exposure, centers at perceiving the reliability of the sources and the trustworthiness of cases they make without knowing either from the prior. In this work, we recognized three basic challenges that have not been particularly tended to in the present truth disclosure composing. The first is "trickiness spread" where a basic number of sources are adding to false claims, making the ID of fair cases troublesome. For example, on Twitter, reports, traps, and effect bots are fundamental instances of sources scheming, either purposely or inadvertently, to spread misrepresentation and obscure reality.

KEYWORDS: Truth Discovery, Social Media, Query, Hadoop Distributed File System

I. INTRODUCTION

In the era of information explosion, data have been woven into every aspect of our lives, and we are continuously generating data through a variety of channels, such as social networks, blogs, discussion forums, crowdsourcing platforms, etc. These data are analyzed at both individual and population levels, by business for aggregating opinions and recommending products, by governments for decision making and security check, and by researchers for discovering new knowledge. In these scenarios, data, even describing the same object or event, can come from a variety of sources. However, the collected information about the same object may conflict with each other due to errors, missing records, typos, out-of-date data, etc. For example, the top search results returned by Google for the query the height of Mount Everest" The second test is "data sparsity" or the "long-tail wonder" where a larger piece of sources simply contributes few cases, giving lacking affirmation to choose those sources' unwavering quality. Third, various present game plans are not adaptable to enormous scale social identifying events because of the concentrated thought of their reality disclosure computations. In this paper, we develop a Truth Discovery plan to address the more than three troubles. In particular, the Truth Discovery contrive together assesses both the unfaltering nature of sources and the legitimacy of cases using a principled approach. We furthermore develop a scattered structure to complete the proposed truth exposure plot using Work Queue in a HT Condor system. The appraisal occurs on three authentic datasets exhibit that the Truth Discovery plot basically beats the best in class truth disclosure procedures to the extent both sufficiency and efficiency. Include "29,035 feet", "29,002 feet" and "29, 029 feet". Among these pieces of noisy information, which one is more trustworthy, or represents the true fact? In this and many more similar problems, it is essential to aggregate noisy information about the same set of objects or events collected from various sources to get true facts. One straightforward approach to eliminate conflicts among multi-source data is to conduct majority voting or averaging. The biggest shortcoming of such voting/averaging approaches is that they assume all the sources are equally reliable. Unfortunately, this assumption may not hold in most cases. Consider the abovementioned "Mount Everest" example: Using majority voting, the result "29,035 feet", which has the highest number of occurrences, will be regarded as the truth. However, in the search results, the information "29; 029 feet" from Wikipedia is the truth. This example reveals that information quality varies a lot among different sources, and the accuracy of aggregated results



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

can be improved by capturing the reliabilities of sources. The challenge is that source reliability is usually unknown a priori in practice and has to be inferred from the data. The sources that provide true information more often will be assigned higher reliability degrees, and the information that is supported by reliable sources will be regarded as truths

II. LITERATURE REVIEW

Onureena Banerjee Laurent El Ghaoui and Alexandred'Aspremont[1] Undirected graphical models offer an approach to portray and clarify the connections among an arrangement of factors, a focal component of multivariate information investigation. The rule of niggardliness manages that we should choose the least difficult graphical model that enough clarifies the information. In this paper we think about pragmatic methods for executing the accompanying way to deal with finding such a model: given an arrangement of information, we take care of a most extreme probability issue with an additional ℓ_1 -standard punishment to make the subsequent diagram as scanty as could be allowed. Numerous creators have contemplated an assortment of related thoughts. In the Gaussian case, display determination includes finding the example of zeros in the backwards covariance network, since these zeros relate to restrictive independencies among the factors. Generally, an avaricious forward-in reverse pursuit calculation is utilized to decide the zero examples (e.g., Lauritzen, 1996). Notwithstanding, this is computationally infeasible for information with even a moderate number of factors. Li and Gui (2005) present an inclination plummet calculation in which they represent the sparsity of the backwards covariance grid by characterizing a misfortune work that is the negative of the log probability work. Speed and Kiiveri (1986) and, all the more as of late, Dahl et al. (Modified 2007) proposed an arrangement of vast scale strategies for issues where a sparsity design for the converse covariance is given and one must gauge the nonzero components of the network.

Peter Bui, Dinesh Rajan, Badi Abdul-Wahid, Jesus Izaguirre and Douglas Thain[2] Today, inquire about researchers confront the test of proficiently what's more, adequately using the wealth of processing assets presently accessible to them through grounds bunches, registering frameworks, and cloud situations. In spite of the fact that there are instruments for building applications for every one of these individual disseminated situations, there are not very many frameworks intended to saddle the registering intensity of these assets at the same time. To address this issue, we created Work Queue, a adaptable ace/laborer system for building vast scale logical outfit applications that range numerous machines counting bunches, frameworks, and mists. Not at all like customary circulated programming frameworks, for example, MPI, Work Queue takes into consideration a versatile specialist pool and in this manner empowers the client to scale the quantity of laborers up or down as required by their application. Moreover, it gives adaptation to internal failure for irregular blunders by effortlessly dealing with specialist disappointments. Besides, Work Queue additionally gives information administration highlights to help information escalated conveyed applicatios.

Xin Luna Dong, Laure Berti-Equille and DiveshSrivastava[3] We are for the most part propelled by incorporating information from the Web. In an assortment of areas, for example, science, business, legislative issues, craftsmanship, excitement, sports, travel, there are a colossal number of information sources that try to give data and a ton of the gave data covers. While a portion of this data is dynamic, a huge bit of the data is about some static part of the world, for example, writers and distributors of books, executives, on-screen characters, and on-screen characters of motion pictures, income of an organization in past years, leaders of a nation previously, and capitals of nations; the information sources once in a while refresh such data. This paper centers on such static data and thinks about a preview of information from various sources. Numerous information sources may reorder, creep, or total information from different sources, and distribute the replicated information without unequivocal attribution. In such applications, thinking about conceivable reliance between sources can regularly prompt more exact truth-revelation results. Preferably, while applying casting a ballot, we might want to overlook replicated data; be that as it may, this raises something like three difficulties. To begin with, in numerous applications we don't know how each source acquires its information, so we need to find copiers from a depiction of information. The revelation is non-minor as sharing basic information does not in itself infer duplicating. Second, notwithstanding when we choose that two sources are reliant,



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

with just a depiction it isn't clear which one is a copier. Third, a copier can likewise give a few information without anyone else or check a portion of the replicated information, so it is improper to overlook all information it gives.

Houping Xiao, Jing Gao, Qi Li, Fenglong Ma and Lu Su YunlongFeng[4] Today, we are living in an information rich world, and the data on a question (e.g., populace/climate/air nature of a specific city) is normally given by various sources. Unavoidably, there exist clashes among the multi-source information because of an assortment of reasons, for example, foundation commotion, equipment quality or malevolent goal to control information, and so forth. An essential inquiry is the means by which to recognize the genuine data (i.e., certainties) among the numerous clashing snippets of data. In light of the volume issue, we can't anticipate that individuals will distinguish truth for each protest physically. In this way, the interest for programmed extraction of realities from clashing multi-source information has taken off as of late.

Richard Farkas, VeronikaVincze, Gyorgy M ora and Janos Csirik GyorgySzarvas[5] Consistently since 1999, the Conference on Computational Natural Language Learning (CoNLL) gives an aggressive shared errand to the Computational Linguistics people group. Following a Multi-year time of multi-dialect semantic job naming and syntactic reliance parsingundertakings, another assignment was presented in 2010, in particular the identification of vulnerability and its etymological extension in regular dialect sentences. The two errands were tended to in the CoNLL-2010 Shared Task, with the end goal to give uniform physically commented on benchmark datasets for both and to analyze their troubles and best in class answers for them. The vulnerability discovery issue comprises of two phases. To start with, catchphrases/signs showing vulnerability ought to be perceived then either a sentence-level choice is made or the semantic extent of the prompt words must be distinguished. The last errand falls inside the extent of semantic investigation of sentences abusing syntactic examples, as support ranges can for the most part be resolved based on syntactic examples subject to the watchword.

Aameek Singh and Ling Liu [6] Decentralized Peer to Peer (P2P) networks offer both opportunities and threats. Its open and decentralized nature makes it extremely susceptible to malicious users spreading harmful content like viruses, trojans or, even just wasting valuable resources of the network. In order to minimize such threats, the use of community-based reputations as trust measurements is fast becoming a de-facto standard. The idea is to dynamically assign each peer a trust rating based on its performance in the network and store it at a suitable place. Any peer wishing to interact with another peer can make an informed decision based on such a rating. An important challenge in managing such trust relationships is to design a protocol to secure the placement and access of these trust ratings. Surprisingly, all the related work in this area either support very limited anonymity or assume anonymity to be an undesired feature and neglect it. In this paper, we motivate the importance of anonymity, especially in such trust based systems. We then present TrustMe a secure and anonymous underlying protocol for trust management. The protocol provides mutual anonymity for both the trust host and the trust querying peer. Through a series of simulation-based experiments, we show that the TrustMe protocol is extremely secure in the face of a variety of possible attacks and present a thorough analysis of the protocol.

III. PROPOSED SYSTEM

Proposed concept deals with providing database by using Hadoop tool we can analyze no limitation of data and simple add number of machines to the cluster and we get results with less time, high throughput and maintenance cost is very less and we are using joins, partitions and bucketing techniques in Hadoop. Advantages in this proposed system is that the problem losing the data is not possible. Here processing of data is efficient. Benefits of this system is that it can process large amount datasets and it consumes less time.

The algorithms that has been used in our work are:

MAPREDUCE ALGORITHM

Generally MapReduce paradigm is based on sending the computer to where the data resides. MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage. In the Map stage the map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 7, Issue 2, February 2019

Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data. In Reduce stage this stage is the combination of the Shuffle stage and the Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

IV. ARCHITECTURE DIAGRAM

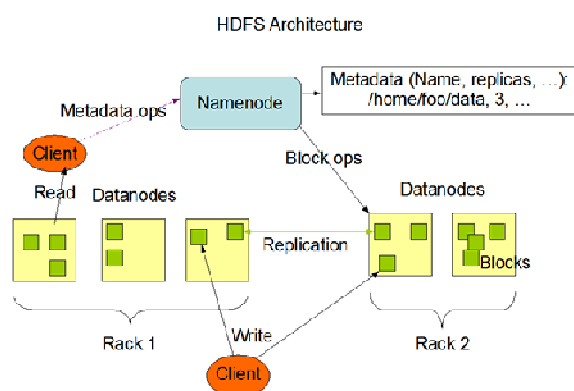


Fig.1 Architecture Diagram

V. MODULES

1. Connector (Sqoop)

Sqoop is a command-line interface application for transferring Truth Discovery data between relational databases (MySQL) and Hadoop. Here in MySQL database having Truth Discovery data have to import it to HDFS using Sqoop. Truth Discovery data can be moved into HDFS/Hive from MySQL and then it will generate the java classes. In previous cases, flow of data was from RDBMs to HDFS. Using "export" tool, we can import data from HDFS to RDBMs. Before performing export, Sqoop fetches table metadata from MySQL database. Thus we first need to create a table with required metadata.

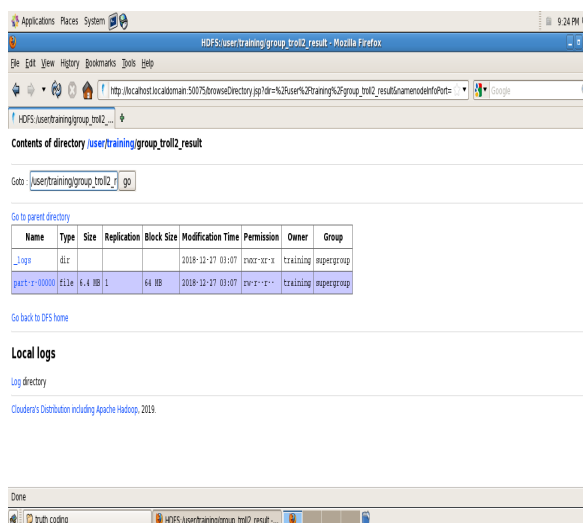


Fig.2 Connector

International Journal of Innovative Research in Computer and Communication Engineering

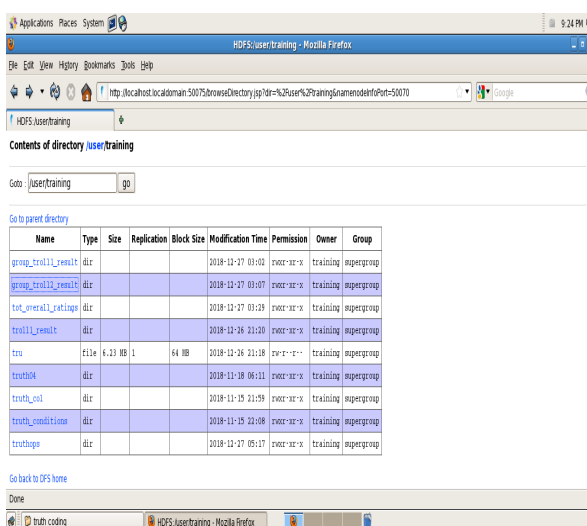
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

2. Analysis Query Language (Hive)

Hive is a data warehouse system for Hadoop that runs SQL like queries called HQL (Hive query language) which gets internally converted to map reduce jobs. In Hive, Truth Discovery data tables and databases are created first and then data is loaded into these tables. Hive as data warehouse designed for managing and querying only structured data that is stored in tables. Hive organizes Truth Discovery data tables into partitions. It is a way of dividing a table into related parts based on the values of partitioned columns. Using partition, it is easy to query a portion of the given dataset. Tables or partitions are sub-divided into buckets, to provide extra structure to the Truth Discovery data that may be used for more efficient querying. Bucketing works based on the value of hash function of some column of a table.



Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
group_trn11_result	dir				2018-11-27 03:00	rwxr-xr-x	training	supergroup
group_trn11_result2	dir				2018-11-27 03:07	rwxr-xr-x	training	supergroup
tbl_order1_xstings	dir				2018-11-27 03:28	rwxr-xr-x	training	supergroup
trn11_result	dir				2018-11-26 23:20	rwxr-xr-x	training	supergroup
tru	file	6.23 KB	1	64 MB	2018-11-26 23:18	rw-r--r--	training	supergroup
truth04	dir				2018-11-18 06:11	rwxr-xr-x	training	supergroup
truth_col	dir				2018-11-15 23:59	rwxr-xr-x	training	supergroup
truth_conditions	dir				2018-11-15 23:08	rwxr-xr-x	training	supergroup
truthops	dir				2018-11-17 05:17	rwxr-xr-x	training	supergroup

Fig.3 Analysis Query Language

3. Analysis Latin Script (Pig)

To analyze Truth Discovery data using Pig, programmers need to write scripts using Pig Latin language and execute them in interactive mode using the Grunt shell. All these scripts are internally converted to Map and Reduce tasks. After invoking the Grunt shell, you can run your Pig scripts in the shell. Except LOAD and STORE, while performing all other operations, Pig Latin statements take a relation as input and produce another relation as output. As soon as you enter a Load statement in the Grunt shell, its semantic checking will be carried out. To see the contents of the schema, you need to use the Dump operator. Only after performing the dump operation, the MapReduce job for loading the data into the file system will be carried out. Pig provides many built-in operators to support data operations like grouping, filters, ordering, etc.

4. Processing (MapReduce)

MapReduce is a framework using which we can write applications to process huge amounts of Truth Discovery data, in parallel, on large clusters of commodity hardware in a reliable manner. MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage. The map or mapper's job is to process the input data. Generally, the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data. This stage is the combination of the Shuffle stage and the Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

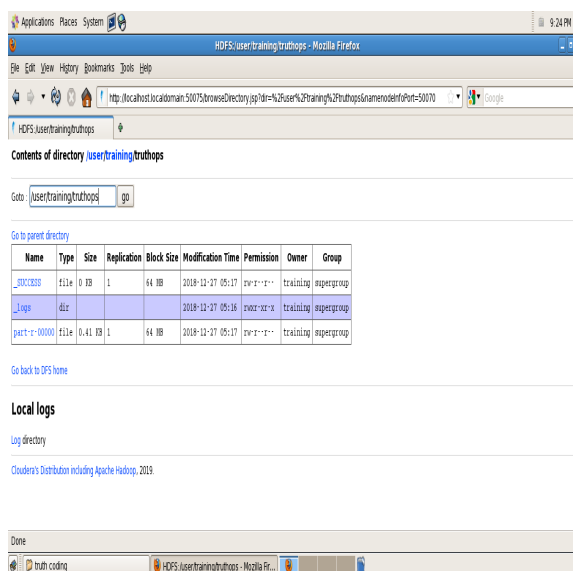


Fig.4 Processing

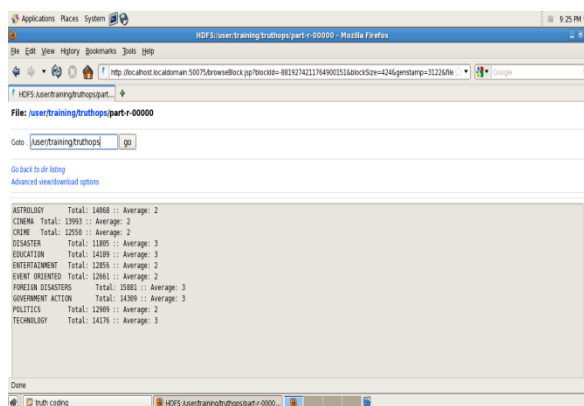


Fig. 5 Discovered result

VI. CONCLUSION

In this paper, we proposed a Truth Discovery framework to address the data veracity challenge in big data social media sensing applications. In our solution, we explicitly considered the source reliability, report credibility, and a source's historical behaviors to effectively address the misinformation spread and data sparsity challenges in the truth discovery problem. To analysis the Truth Discovery data in Hadoop ecosystem to improve the performance based on over all Truth Discovery details. Hence by using Hadoop tool faster and efficiently processing the data.

VII. FUTURE ENHANCEMENTS

Apache Spark is an open source processing engine built around speed, ease of use, and analytics. If you have large amounts of data that requires low latency processing that a typical Map Reduce program cannot provide, Spark is



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 2, February 2019

the alternative. Spark provides in-memory cluster computing for lightning fast speed and supports Java, Scala, and Python APIs for ease of development.

REFERENCES

- [1] O. Banerjee, L. E. Ghaoui, and A. dAspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine learning research*, 9(Mar):485–516, 2008.
- [2] S. Bhuta and U. Doshi. A review of techniques for sentiment analysis of twitter data. In *Proc. Int Issues and Challenges in Intelligent Computing Techniques (ICICT) Conf*, pages 583–591, Feb. 2014.
- [3] J. Bian, Y. Yang, H. Zhang, and T.-S. Chua. Multimedia summarization for social events in microblog stream. *IEEE Transactions on multimedia*, 17(2):216–228, 2015.
- [4] P. Bui, D. Rajan, B. Abdul-Wahid, J. Izaguirre, and D. Thain. Work queue+ python: A framework for scalable scientific ensemble applications. In *Workshop on python for high performance and scientific computing at sc11*, 2011.
- [5] P.-T. Chen, F. Chen, and Z. Qian. Road traffic congestion monitoring in social media with hinge-loss markov random fields. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 80–89. IEEE, 2014.
- [6] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. In *Proceedings of the VLDB Endowment*, pages 550–561, 2009.
- [7] X. X. et al. Towards confidence in the truth: A bootstrapping based truth discovery approach. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016.
- [8] R. Farkas, V. Vincze, G. Mora, J. Csirik, and G. Szarvas. The conll-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 2010.
- [9] R. Feldman and M. Taquq. *A practical guide to heavy tails: statistical techniques and applications*. Springer Science & Business Media, 1998.
- [10] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *Proc. of the ACM International Conference on Web Search and Data Mining (WSDM'10)*, pages 131–140, 2010.
- [11] H. Hu, G. J. Ahn, and J. Jorgensen. “Multiparty access control for online social networks: Model and mechanisms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1614–1627, July 2013.
- [12] N. Vishwamitra, Y. Li, K. Wang, H. Hu, K. Caine, and G.-J. Ahn, “Towards pii-based multiparty access control for photo sharing in online social networks,” in *Proceedings of the 22Nd ACM on Symposium on Access Control Models and Technologies*, June 2017, pp. 155–166.
- [13] P. Mehregan and P. W. Fong, “Policy negotiation for co-owned resources in relationship-based access control,” in *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies*, June 2016, pp. 125–136.
- [14] J. Golbeck, “Trust on the world wide web: A survey,” *Foundations and Trends in Web Science*, vol. 1, no. 2, pp. 131–197, 2008.
- [15] S. Zakhary, M. Radenkovic, and A. Benslimane, “Efficient location privacy-aware forwarding in opportunistic mobile networks,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 893–906, February 2014.