International Journal of Web-Based Learning and Teaching Technologies
Volume 15 • Issue 2 • April-June 2020

Topic Sensitive User Clustering Using Sentiment Score and Similarity Measures:
Big Data and Social Network

Bharat Tidke, SVNIT, Surat, India
Rupa Mehta, SVNIT, Surat, India
Dipti Rana, SVNIT, Surat, India
Hullash Jangir, SVNIT, Surat, India

ABSTRACT

Social media data (SMD) is driven by statistical and analytical technologies to obtain information for various decisions. SMD is vast and evolutionary in nature which makes traditional data warehouses ill suited. The research aims to propose and implement novel framework that analyze tweets data from online social networking site (OSN; i.e., Twitter). The authors fetch streaming tweets from Twitter API using Apache Flume to detect clusters of users having similar sentiment. Proposed approach utilizes scalable and fault tolerant system (i.e., Hadoop) that typically harness HDFS for data storage and map-reduce paradigm for data processing. Apache Hive is used to work on top of Hadoop for querying data. The experiments are performed to test the scalability of proposed framework by examining various sizes of data. The authors' goal is to handle big social data effectively using cost-effective tools for fetching as well as querying unstructured data and algorithms for analysing scalable, uninterrupted data streams with finite memory and resources.

KEYWORDS

Big Data, Big Social Data, Sentiment Analysis, Stream Mining

INTRODUCTION

In recent scenario the modern social media, mobile or web strategy involved in communication has been technology concentric that makes data to grow rapidly, ultimately creates large noisy and unstructured data. This gave sudden rise (Felt, 2016) to concept of Bigdata. (Kitchin, 2014) describe Bigdata by 3 V's: large volume, uninterruptable velocity, different data structure as variety which can be extensive in opportunity, interactive in nature, and springy in quality. Many theories in social science like correlation has been proven to be pertinent to social media. As per social correlation theory (Tang, Tan, & Liu, 2014), contiguous users in a social media have similar behaviors or attributes. These phenomena clarify user's inclination to connect or follow with others having certain similarity or sharing the same surroundings. The quantity of information available for harnessing in social media is massive and growing every second. Increasing volumes of data (Tang et al., 2014), has been a major challenge for the data oriented companies like Google, Yahoo, LinkedIn, Twitter, and

DOI: 10.4018/IJWLTT.2020040103
This article, originally published under IGI Global's copyright on April 1, 2020 will proceed with publication as an Open Access article starting on January 28, 2021 in the gold Open Access journal, International Journal of Web-Based Learning and Teaching Technologies (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Facebook for which different solutions are proposed and implemented. Managing the voluminous and evolutionary real-world data demands the scalable data management system. Emerging distributed storage system like Hadoop, NoSQL and Cloud based infrastructure aids to reduce the cost for data storage. Researchers and practitioners (Beyer & Laney, 2012; Chen & Zhang, 2014; Wang, Kung, & Byrd, 2018) work dedicated to such huge volume of data shows constant growing interest. Similarly Open source software communities like Apache comes with opinion that due to massive increase in size of dataset it has been quiet difficult to acquire, store and analysed such large volume of data.

Social media (Zafarani, Abbasi, & Liu, 2014) is indeed a way to communicate virtually, in terms of opinions and sentiments of people that can be used by businesses and governments organisations to act accordingly. The process of collecting, integrating, storing and processing of Big social media data to gain information is highly tedious task which yet to be solved fully. In addition, such data needs pre-processing as it contains outliers and noisy data. Similarly, post, opinions or replies (Tang, Tan, & Liu, 2009) from various user on same or different topics have sentiments attached to it. Twitter is a microblogging sites which become one of the main platforms for capturing data to do further analysis. These analysis can be useful for finding out polarity in terms of positive or negative (Tang et al., 2009), detecting trends (Alsaedi, Burnap, & Rana, 2017; Lambrecht, Tucker, & Wiertz, 2018), community detections (Wen et al., 2017), recommendation of product and services (Abbas, Zhang, & Khan, 2015). This paper mainly focuses on stream data management and data analytics for finding similarity and sentiment analysis of user using tweets from Twitter data. Also, investigates and implement various imminent technologies for acquiring, storing and analysing Big social media data. The contributions of proposed work are highlighted below:

- Configured a highly reliable system i.e. Hadoop to store very large files in distributed environment. To ingest data as stream we configure Apache Flume to fetch Twitter data;
- To store Twitter data in structure format, we need to pre-process by storing data in Hive tables. On Hive table we implement a process to calculate sentiments of tweets using HiveQL based on AFFINE Dictionary;
- Proposed architecture for extracting information from large number of tweets to cluster similar user. For calculating Similarity between Tweets, we designed a MapReduce process for efficiency and fault-tolerance which uses text mining approach such as TF-IDF and cosine similarity measure to calculate values for similar tweets and users;
- Finally, results generates output clusters based on sentiments and similarity score.

RELATED WORK

The fast growth of social media networks (Tang et al., 2009) permit various users to relate, which helps to form a group of people who are eager to interact, share, and collaborate using social media platforms. Analysing social media is a tedious task and the existing approaches as well as methods needs to adapt and integrate them to emerging Bigdata models (Chen & Zhang, 2014) for enormous storage as well as processing. Various paradigm like Apache Hadoop and Spark comes into existence that makes possible to have scalable and distributed application of ML algorithms in diverse fields. These Bigdata paradigms consist of numerous in built libraries to improve performance of existing techniques and algorithms (Beyer & Laney, 2012; Chen & Zhang, 2014).

Social Media Mining

Social media mining has been divided into three categories [Figure 1], i.e. user based, link based and content based (Etter, Colleoni, Illia, Meggiorin, & D'Eugenio, 2018; Dridi & Recupero, 2017).

User based techniques explore behaviour modelling and build feature patterns from particular social usernames, idiom and linguistics. This information can be leverage for user classification,

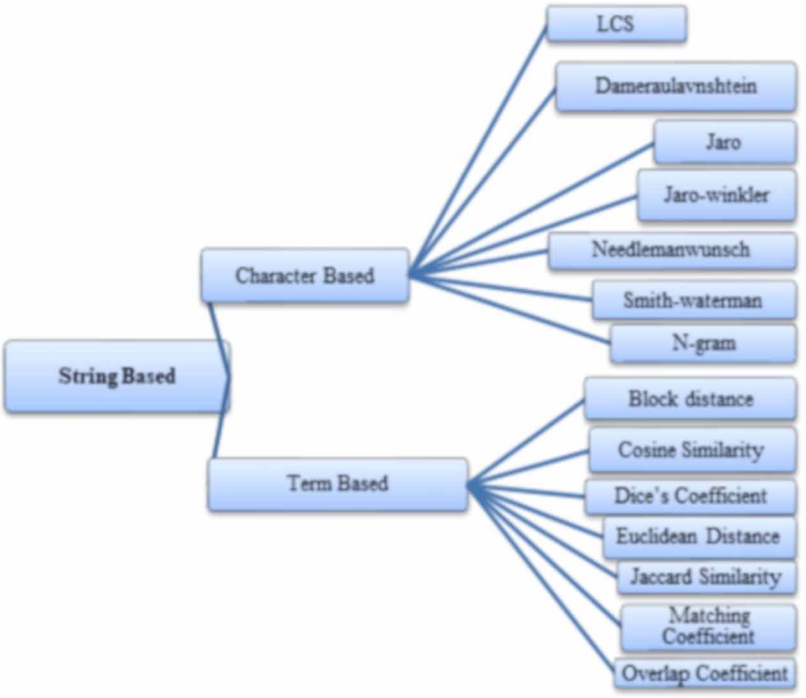
Figure 1. Various techniques for social media mining

spammers and many more. Relation based structural analysis performs linkage analysis by exploiting structure features in social circles. This linkage data can use for finding node centrality (Peng, Wang, & Xie, 2017), social ties, link prediction (Bliss, Frank, Danforth, & Dodds, 2014), community detection (Liu 2012; Sapountzi & Psannis, 2016). The unstructured heterogeneous contents bind to the nodes and edges of a social network have compelling importance in finding patterns and information in Big social network (Bello-Orgaz, Jung, & Camacho, 2016).

Content-based techniques (Bello-Orgaz et al., 2016) fits in environment of Big social data analytics, as data is vast and unstructured. Sentiment Analysis using various contents such as tweets, post etc. becomes one of most research area in social media mining that enables us to find out polarity about sentences, documents and aspect of particular sentence or document in terms of negative or positive. Sentiment analysis (Liu, 2012) can be categorized into first statistical based in which machine learning techniques can be used to give broad view of any document or sentence for prediction or recommendation. Second is lexicon based mainly focusing on abstract view and can use dictionaries or any available annotation for finding polarity and final method is hybrid based on these two methods. Similarly, lots of research has been done on text similarity techniques (Sun, Ma. & Wang, 2015; Yu, Li, Deng, & Feng, 2016) which shows it has vital role in the fields such as summarization, text mining, information retrieval, spam mail detection etc. Text similarity techniques basically divided into two categories which are Lexical similarity and Semantic similarity. Lexical method (Pradhan, Gyanchandani, & Wadhvani, 2015) mostly used various string based algorithm [Figure 2]. String based similarity methods performs different operations based on character structure and manner in string comes in sequence.

Among many recent contributions in social Bigdata (Jansen, Zhang, Sobel, & Chowdury, 2009) explained and compared different classification and manual coding techniques approach on users brand sentiments using Twitter data and they found out that social networking sites like Twitter is a platform where user can communicate with various commercial businesses. (Trattner & Kappe, 2013) suggested opinions and sentiments can be useful for purchasing online products as they performed several experiments on Facebook stream Bigdata. (Ma, Yang, Lyu, &, 2008) performed sentiment analysis using three different models based on heat diffusion process and comes up with opinion that same can be applied to Big social network data analytics also in scalable manner. Social media data

Figure 2. Various string based similarity techniques



(Chen, Hsu, & Lee, 2013; Bohlouli, Dalter, Dornhofer, Zenkert, & Fathi, 2015; He, Wu, Yan, Akula, & Shen, 2015) can improve the quality of new products based on customer opinions or reviews.

PROPOSED APPROACH

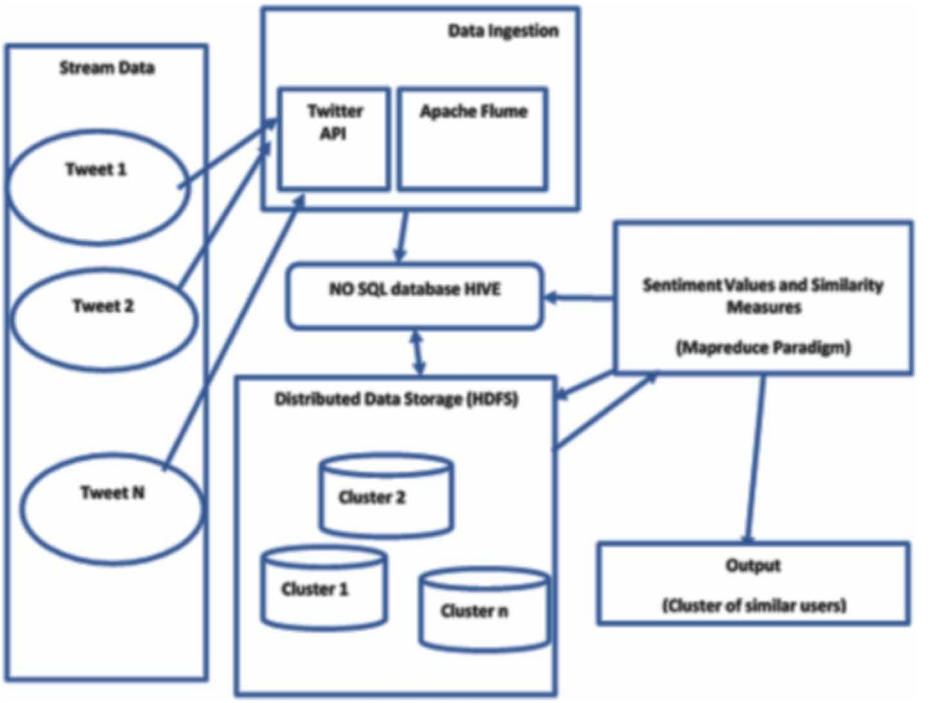
Our goal is to analyse Twitter stream data to find out highly related Tweets and similar users to detect community, patterns and behaviour among different users. To achieve desired results, an architecture to extract information from social Bigdata has been proposed and implemented [Figure 3]. Firstly, Apache Flume an open source framework is used ingest tweet as stream from Tweeter API randomly using specified keyword or topic. Next, we integrated Apache Flume and Hadoop distributed storage platform (Hadoop HDFS) for storing ingested unstructured stream tweet data to achieve scalability. Sentiments of Tweets are calculated using Hive based on AFINE dictionary which contains collections of words having sentiment score attached in between -5 to +5. Similarly, for measuring similarity, stored data has been process using Mapreduce for efficiency and fault-tolerance. Finally, results gives a group of highly related Tweets and similar users.

Similarity Measures Model

Tweets are normally textual data in JSON format. Text data has been used for finding similarity between the tweets. So, existing text similarity techniques can be useful to extract similar sentences and word for further analysis.

To extract similar tweets there is need to understand the linguistics of users. Due to lack of proper grammar in tweets we focused mainly on keywords extraction which is an important technique in information retrieval. There are variety of challenging task for NLP in finding keywords and to extract summary of tweets like NRE (Name Entity Recognition), stemming

Figure 3. Proposed architecture for finding sentiment and similarity from Twitter data



negation handing (Cambria, Schuller, Xia, & Havasi, 2013). However, mining of sentiment only limited to recognise positive or negative significance of the document or sentences (Benedetti, Beneventano & Bergamaschi, 2016). To overcome this challenge we pre-process data using following criteria:

- **Transformation:** This process converts raw tweets which are in JSON format into semi-structure format using Apache Hive;
- **Filtration:** Eliminating irrelevant words from tweets such as common Adverb, Pronouns, Conjunction, Prepositions, Non-informative we consider the remaining highly relevant words for further process. All irrelevant words are stored in one dictionary (stop word list). Proposed framework evaluates the data of incoming tweets with dictionary and removes irrelevant word which are same;
- **Stemming:** After removing irrelevant words next we perform stemming process that discards prefixes or suffixes of the words and construct a lexicon for speedy stemming process. The next process is to calculate the occurrence of each word in every tweets to give weights to each keyword.

To summarize tweets using few keywords we selected TF/IDF technique which chooses the most frequently occurring terms (*tf*). The challenge lies giving weightage to individual words as the most of recurrent word occur quiet frequently in all sentences or documents, is of little use. Hence, a metric (inverse document frequency or idf) is needed to extract uniqueness of each word i.e. how intermittently the word comes in all documents. Therefore, the multiplication of tf*idf of individual word gives the weightage to this word. Words with a better tf-idf score comes recurrently and gives the useful statistics and information about individual tweets.

- **Word count:** In this process finding words which have some meaning and indicate information about opinion sentences. Mostly these opinion words are adjectives but finding out such word there is need of understanding number of count of word in sentences.

Algorithm 1: Word Count (To count the existence of individual term (t) in each Tweet)

Let: Term- (t), TweetId- Tid, MP- Mapper, Rd- Reducer, Sum- S, Cosine Similarity - CS

1: class MP
2: procedure Map $(Tweet)$
3: ***for*** $each(t) \in Tweet$
4: $mark(((t),Tid),1)$
5: class Rd
6: procedure Reduce $(((t)\ ,Tid)\ ,K[1\ ,1\ ,\ldots\ ,n]\ )$
7: $A=0$
8: ***for*** $each\ n \in K\ do$
9: $S=S+1$
10: ***return*** $(((t),Tid),K)$

Term frequency: In this process to find out frequency of word in whole tweets is calculated of opinion words.

Algorithm 2: Term Frequency (Total count (C) of (t) for every Tweet is calculated)

1: class Mp
2: procedure $Map(((term,Tid),C)$
3: ***for*** $each\ item \in (((t),Tid)$
4: $write(Tid,((t),C))$
5: class Rd
6: $procedure\ Reduce(Tid,((t),C))$
7: $K=0$
8: ***for*** $each tuple \in (((t),C)\ do$
9: $K=K+C$
10: ***return*** $((Tid,N),((t),C))$

Algorithm 3: $Tf-Idf$ ($Tf-Idf$ for every (t) in a Tweet is calculated)

1: class Mp
2: procedure $Map(((Tid,K),((t),C))$
3: ***for*** every item ∈ ((t), C)
4: mark ((t), (Tid, o, N))

$$tf-idf=\frac{n}{K}\frac{|Nt|}{|\{d \epsilon Nt : t \epsilon d\}|}$$

Where $\left|Nt\right|$ is total Tweets in dataset and $\left|\left\{d \in Nt : t \in d\right\}\right|$ number of Tweet where $t-\left(t\right)$ appears.

```
5: class Rd
6:  procedure Reduce((t),(Tid,C,K))
7:  n=0
8:  for each item∈(Tid,C,K) do
9:  n=n+1
10:  tf=C /K
11: idf=log(|T| /(1+n))
12: return(Tid,((t) ,tf × idf))
```

The output summarized tweets are further attached their Sentiments value using AFINN dictionary

In the 4th stage pairwise probable correlation of two Tweets are given and cosine value for individual is calculated. Consider k Tweets in the dataset, a correspondence similarity matrix of given set is created as:

$$\binom{k}{2} = \frac{k!}{2!\left(k-2\right)!}$$

The cosine angle between two tweets or set of tweets are computed which gives the cosine similarity between two tweets or set of tweets. This measure evaluates of positioning and not magnitude.

Algorithm 4: To compute CS

```
1: class Mp
2:  procedure Map(Tweets)
3:  k=length of tweet
4:  for i=0 to k
5:  for j=i+1 to k
6:  fetch((Tweets[i].id,Tweets[j].id),(Tweets[i].tfidf,Tweets[j].tfidf))
7: class Rd
8:  procedure Reduce((Tid_X,Tid_Y),(TweetX.tfidf,TweetY.tfidf))
9: X=TweetX.tfidf
10: Y=TweetY.tfidf
11:  Cosine =S(X × Y) (√(sum(X2)) × sqrt(sum(Y2)) )
12:  return ((Tid _X,Tid _Y),cosine)
```

We need to find out highly related Tweets, to achieve this goal steps are listed as follows:

- We counted Sentiments of Tweets using AFINN dictionary;
- Also calculated Similarity of Tweets by Cosine similarity;
- Then we will apply the given algorithm to find out highly related Tweets.

Algorithm 5: Most_related Tweets

```
1: read sentiment.txt file
2: read similarity.txt file
3: store sentiment file up to Null in Hash map as
tempStr=str.split("\ t")
if(tempStr.length==2)
sentimentMap.put(Long.parseLong(tempStr[0]),tempStr[1])
4: store similarity file up to Null in Hash map also as
tempStr=str.split(" ",2 )
if (tempStr[1].length() !=0)
svm4MapOne.put(Long.parseLong(tempStr[0]),tempStr[1])
5:  for i = 0 to N in svmMapOne
for loop until String strTmp :firstEntry.getValue().split(" ")
rightIDList.add(strTmp.substring(0,18))
                rightSimValList.add(strTmp.substring(19))
String tmp=sentiment value of Tweet ID
String tmp1=sentiment value of Tweet ID
prevDiff=Math.abs(tmp -tmp1)
for loop until all ID to right side of ID's to similarity file
String tmp=sentiment value of Tweet ID
String tmp1=sentiment value of Tweet ID
currDiff=Math.abs(tmp -tmp1)
if prevDiff is equal to currDiff
then write ID and its corresponding similarity value
6 :repeat step 5 for other Tweet IDs
7 :end.
```

Similarity and sentiment may possibly be measured as classification techniques for textual data, especially when dealing with social media data having post and tweets etc. But their functions are distinct. Since sentiment analysis goal is to categorize into opinions for example 'positive' and 'negative'. While a similarity measure classifies sentences or documents based on their similar features vectors. In case of textual data it may be keywords or whole document. This would give broader view about what the post or tweets are about, not only positive or negative. In our approach we combined both similarity as well as sentiment techniques to gain insight of user tweets so that they can cluster better.

IMPLEMENTATION METHODOLOGY

In our implementation for collecting tweet data, API provided by Twitter has been used. In that keyword related to any topic is given as input to fetch relevant tweets. For example in our method we used "Tobacco" as keyword. We collected three datasets as a stream tweet of size 560 MB, 1 GB and 1.5 GB. We used Apache Flume for data ingestion. To store and process data in distributed manner we created Hadoop cluster of 5 machines where 4 machines are slaves nodes and 1 is master node.

Apache Hive used to analyze Sentiment of Tweets using AFINN dictionary and to get Similarity between Tweets we implemented java program in Mapreduce programming model.

Apache Flume

Apache Flume is configured to ingest stream Tweet data in that we configure the Twitter as a source, 1% Firehose Memory channel. In addition, HDFS sink is configured in conf folder of Apache Flume. To fetch data from the Twitter API the authentication in the form of keys is needed such as Consumer key etc.

Apache Hive

Apache Hive is used to process data to store in tables. At first stage we designed an Avro schema file as TwitterAvroSchema.avsc to parse Jason file of Tweet data. Then this file is load Tweet data in internal table to further utilization of table where Flume's data is loaded. To retrieve and to analyze Tweet data Hive External Tables are created.

Hadoop

The fundamental consideration for storing and processing data using Hadoop (HDFS and MapReduce) is motivated by both continuously increasing sizable data and expense of computational hardware. The main task of Hadoop is to control the commodity machines for outsized scale of storing as well as computational capability to handle load, alternatively achievable by several costly workstation computers. The benefit to implement our approach on Hadoop platform is that "Hadoop is no cost and open source" platform. HDFS is mainly used as a distributed storage and Mapreduce normally for parallel processing task. Full functionality of Hadoop can be find out on www.hadoop.apache.org.

Performance Result and Analysis

This section provides the analysis of sentiment value and similarity measures using cosine similarity metrics. Twitter API is used for stream Tweet ingestion from Twitter site using Apache Flume which is stored in HDFS file. We calculated and combined Sentiments of Tweets on Hive table data using a dictionary called AFINN having of 2500 unique terms graded from +5 to -5 depending on their sentiments and similarity value with threshold 0.75 using proposed method to find similar tweets and users, some of observing examples [Table 1 and Table 2]. We also tested our approach in centralizes as well distribute environment i.e. Hadoop.

The proposed approach consists of many stages beginning from data ingestion, storage, to find sentiment value for each individual tweet and to get similar users, tweets using cosine similarity. Every stage used individual execution time and programming model like Hive SQL for Sentiment value calculation and MapReduce for similarity measures. We consider time complexity for similarity measure and plot relationships between size of data with respect to time taken to find out users and

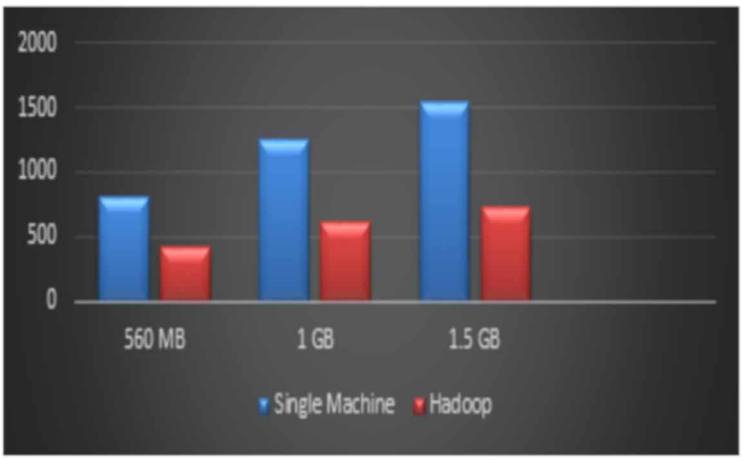
Table 1. Examples for Tweets with its sentiments value

Sr. No	Id	Tweets
1.	737614966610481154	Its a amazing day to quit smoking and tell others as well World No Tobacco Day Help Him Choose Life nicotex in (sentiment value=4.0)
2.	73760866612414464	Tobacco companies kill their best customers (sentiment value=2)
3.	737615165605040128	World No Tobacco Day It should be world no tobacco year Why do people pay to get killed (sentiment value=-1.6)
4.	737591336124088320	2010 used to smoke nearly 30 ciggs a day couldnt run for shit (sentiment value=-4.0)

Table 2. Sample of Tweet ID with similar Tweet ID and cosine similarity

Tweet ID(Sentiment)	Similar Tweet IDs: Cosine Similar Value(Sentiment)
737614966610481154(4.0)	737608180339867651:0.93529628863057(3.0) 737602960406908932:0.9371144658433982(3.0) 737613674639020032:0.9644933285995672(4.0)
737602189149900800(2.3)	737598497071857665:0.752314667448317(2.5) 737598662239358976:0.7338862272944054(2.5) 737598552281481216:0.7143935599578113(2.5) 737598594417496067:0.7238446092620383(2.5) 737601548449021954:0.7034720274783445(2.5)

Figure 4. Time comparison of centralized vs. Distributed environment in terms of seconds to calculate similar tweets



tweets using cosine similarity for particular tweet id. We compare two system i.e. centralized single machine and Hadoop distributed system cluster consist of 5 commodity machines. If the size of data increases from 560 MB to 1.5 GB time taken by single machine centralized system increase drastically. Hadoop distributed system outperforms distributed environment centralized single machine in terms of execution time [Figure 4].

CONCLUSION AND FUTURE WORK

Bigdata requires novel techniques, algorithms to proficiently extract huge and unstructured data in real time. Bigdata techniques are determined by particular applications. Social media expansion and range have greatly affected the way of communication and knowledge interpretation. In our approach, focused is on social network data in the form of tweets from Twitter data. A novel architecture has been proposed and implemented for user clustering based on stream data using sentiment value and similarity measures. This method can be used for many applications like community detection, behaviour similarity detections and group recommendation. In future, we are planning to compare our approach with other similar approaches for mining Twitter data. In addition, test proposed approach on real-time systems like Apache Spark and Apache Storm as it mainly processes data using in-memory and distributed computing technologies.

REFERENCES

Abbas, A., Zhang, L., & Khan, S. U. (2015). A survey on context-aware recommender systems based on computational intelligence techniques. *Computing*, *97*(7), 667–690. doi:10.1007/s00607-015-0448-7

Alsaedi, N., Burnap, P., & Rana, O. (2017). Can we predict a riot? Disruptive event detection using twitter. *ACM Transactions on Internet Technology*, *17*(2), 18. doi:10.1145/2996183

Bello-Orgaz, G., Jung, J. J., & Camacho, D. (2016). Social big data: Recent achievements and new challenges. *Information Fusion*, *28*, 45–59. doi:10.1016/j.inffus.2015.08.005

Benedetti, F., Beneventano, D., & Bergamaschi, S. (2016). Context Semantic Analysis: a knowledge-based technique for computing inter-document similarity. In *International Conference on Similarity Search and Applications* (pp. 164-178). Springer. doi:10.1007/978-3-319-46759-7_13

Beyer, M. A., & Laney, D. (2012). *The importance of 'big data': a definition*. Stamford, CT: Gartner.

Bliss, C. A., Frank, M. R., Danforth, C. M., & Dodds, P. S. (2014). An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science*, *5*(5), 750–764. doi:10.1016/j.jocs.2014.01.003

Bohlouli, M., Dalter, J., Dornhöfer, M., Zenkert, J., & Fathi, M. (2015). Knowledge discovery from social media using big data-provided sentiment analysis (SoMABiT). *Journal of Information Science*, *41*(6), 779–798. doi:10.1177/0165551515602846

Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, *28*(2), 15–21. doi:10.1109/MIS.2013.30

Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*, 314–347. doi:10.1016/j.ins.2014.01.015

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, *19*(2), 171–209. doi:10.1007/s11036-013-0489-0

Chen, W., Hsu, W., & Lee, M. L. (2013). Making recommendations from multiple domains. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 892-900). ACM.

Dridi, A., & Recupero, D. R. (2017). Leveraging semantics for sentiment polarity detection in social media. *International Journal of Machine Learning and Cybernetics*, 1–11.

Etter, M., Colleoni, E., Illia, L., Meggiorin, K., & D'Eugenio, A. (2018). Measuring organizational legitimacy in social media: Assessing citizens' judgments with sentiment analysis. *Business & Society*, *57*(1), 60–97. doi:10.1177/0007650316683926

Felt, M. (2016). Social media and the social sciences: How researchers employ Big Data analytics. *Big Data & Society*, *3*(1), 2053951716645828. doi:10.1177/2053951716645828

He, W., Wu, H., Yan, G., Akula, V., & Shen, J. (2015). A novel social media competitive analytics framework with sentiment benchmarks. *Information & Management*, *52*(7), 801–812. doi:10.1016/j.im.2015.04.006

Jansen, B. J., Zhang, M., Sobel, K., & Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, *60*(11), 2169–2188. doi:10.1002/asi.21149

Kitchin, R. (2014). The data revolution: Big data, open data, data infrastructures and their consequences. *Sage (Atlanta, Ga.)*.

Lambrecht, A., Tucker, C., & Wiertz, C. (2018). Advertising to early trend propagators: Evidence from twitter. *Marketing Science*, *37*(2), 177–199. doi:10.1287/mksc.2017.1062

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies, 5*(1), 1-167.

Ma, H., Yang, H., Lyu, M. R., & King, I. (2008). Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 233-242). ACM. doi:10.1145/1458082.1458115

Peng, S., Wang, G., & Xie, D. (2017). Social Influence Analysis in Social Networking Big Data: Opportunities and Challenges. *IEEE Network*, *31*(1), 11–17. doi:10.1109/MNET.2016.1500104NM

Pradhan, N., Gyanchandani, M., & Wadhvani, R. (2015). A Review on Text Similarity Technique used in IR and its Application. *International Journal of Computers and Applications*, *120*(9).

Sapountzi, A., & Psannis, K. E. (2016). Social networking data analysis tools & challenges. *Future Generation Computer Systems*.

Sun, Y., Ma, L., & Wang, S. (2015). A comparative evaluation of string similarity metrics for ontology alignment. *Journal of Information and Computational Science*, *12*(3), 957–964. doi:10.12733/jics20105420

Tang, H., Tan, S., & Cheng, X. (2009). A survey on sentiment detection of reviews. *Expert Systems with Applications*, *36*(7), 10760–10773. doi:10.1016/j.eswa.2009.02.063

Tang, J., Chang, Y., & Liu, H. (2014). Mining social media with social theories: A survey. *ACM SIGKDD Explorations Newsletter*, *15*(2), 20–29. doi:10.1145/2641190.2641195

Trattner, C., & Kappe, F. (2013). Social stream marketing on Facebook: A case study. *International Journal of Social and Humanistic Computing*, *2*(1-2), 86–103. doi:10.1504/IJSHC.2013.053268

Wang, Y., Kung, L., & Byrd, T. A. (2018). Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological Forecasting and Social Change*, *126*, 3–13. doi:10.1016/j.techfore.2015.12.019

Wen, X., Chen, W. N., Lin, Y., Gu, T., Zhang, H., Li, Y., & Zhang, J. (2017). A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. *IEEE Transactions on Evolutionary Computation*, *21*(3), 363–377.

Yu, M., Li, G., Deng, D., & Feng, J. (2016). String similarity search and join: A survey. *Frontiers of Computer Science*, *10*(3), 399–417. doi:10.1007/s11704-015-5900-5

Zafarani, R., Abbasi, M. A., & Liu, H. (2014). *Social media mining: an introduction*. Cambridge University Press. doi:10.1017/CBO9781139088510

Bharat Tidke obtained his M.Tech degree in Computer Engineering from Sardar Vallabhbhai National Institute of Technology, Surat, India. He is currently pursuing PhD from Sardar Vallabhbhai National Institute of Technology, Surat, India. He published many papers in reputed international journals and conferences His interests include soft computing, Big data, Machine Learning and Social Network data analytics.

Rupa Mehta is an Associate professor in Department of Computer Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, India. She completed her M.Tech (Research) and Ph.D from Sardar Vallabhbhai National Institute of Technology, Surat. She has several years of experience in teaching and research, also published many papers in reputed international journals and conferences. Her area of research includes Data Mining, Big Data and Artificial Intelligence.

Dipti Rana is an assistant professor in Department of Computer Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, India. She completed her M.Tech (Research) and Ph.D. from Sardar Vallabhbhai National Institute of Technology, Surat. She has several years of experience in teaching and research, also published many papers in reputed international journals and conferences. Her area of research includes DBMS, Web Data Mining, Pattern Recognition, Prediction Mining and Big Data.

Hullash Jangir is pursuing M.Tech degree in Computer Engineering from Sardar Vallabhbhai National Institute of Technology, Surat. His area of interest includes Data Mining and Big Data.