International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)

http://www.ijcsjournal.com Volume 3, Issue 1, No 5, 2015. ISSN: 2348-6600
Reference ID: IJCS-091 PAGE NO: 520-524.

SENTIMENTAL ANALYSIS WITH BIGDATA USING HADOOP

Stud. C. BalajiKiran [#1] Prof. A.T. Ravi [#2]

[#1 2] *Department of Computer Science, SSM College of Engineering, Komarapalayam, Namakkal, Tamilnadu, India.*
[1] balajikiran@live.com [2] atravin@gmail.com

***Abstract*— Big data is the term for a collection of data sets which are large and complex, it contain structured and unstructured both type of data. Data comes from everywhere, sensors used to gather climate information, posts to social media sites, digital pictures and videos etc. This data is known as big data. The proliferation of textual data in business is overwhelming. Unstructured textual data is being constantly generated via call center logs, emails, documents on the web, blogs, tweets, customer comments, customer reviews, and so on. While the amount of textual data is increasing rapidly, businesses' ability to summarize, understand, and make sense of such data for making better business decisions remain challenging. This paper takes a quick look at how to organize and analyze textual data for extracting insightful customer intelligence from a large collection of documents and for using such information to improve business operations and performance.**

Index Terms— Big Data, Hadoop, Sentimental Analysis ***(Key words)***

I. INTRODUCTION

The origin of the term 'Big Data' is due to the fact that we are creating a huge amount of data every day. Usama Fayyad in his invited talk at the KDD BigMine'12Workshop presented amazing data numbers about internet usage, among them the following: each day Google has more than 1 billion queries per day, Twitter has more than 250 million tweets per day, Facebook has more than 800 million updates per day, and YouTube has more than 4 billion views per day. The data produced nowadays is estimated in the order of zettabytes, and it is growing around 40% every year. A new large source of data is going to be generated from mobile devices and big companies as Google, Apple, Facebook and Yahoo are starting to look carefully to this data to find useful patterns to improve user experience. "Big data" is pervasive, and yet still the notion engenders confusion .The opportunity cost of any business to ignore unstructured data is paramount in today's fierce competitive world. According to a survey, unstructured data takes a lion's share in digital space and approximately occupies 80% by volume compared to only 20 for structured data. While the unstructured data is available in abundance, the number of software products and solutions that can accurately analyze the text, present insights in an understandable manner along with the ability to integrate such insights readily into other extant models that use numerical only data are rare. Machine learning algorithms designed to analyze numerical data exactly know the structure of numbers and they are pre-programmed to process the data with precision. In case of natural language, it gets very problematic. Dialects, jargon, misspellings, short forms, acronyms, colloquialism, grammatical complexities, mixing one or more languages in the same text are just some of the fundamental problems unstructured data poses. It makes extremely difficult to precisely analyze unstructured data in the same way we process structured data.

II. TYPES OF BIG DATA AND ITS SOURCES

There are two types of big data: structured and unstructured.

Structured data are numbers and words that can be easily categorized and analyzed. These data are generated by things like network sensors embedded in electronic devices, smart phones, and global positioning system (GPS) devices. Structured data also include things like sales figures, account balances, and transaction data.

Unstructured data include more complex information, such as customer reviews from commercial websites, photos and other multimedia, and comments on social networking sites. These data cannot easily be separated into categories or analyzed numerically. "It uses natural language."Analysis of unstructured data relies on keywords, which allow users to filter the data based on searchable

International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)
http://www.ijcsjournal.com Volume 3, Issue 1, No 5, 2015. ISSN: 2348-6600
Reference ID: IJCS-091 PAGE NO: 520-524.

terms. The explosive growth of the Internet in recent years means that the variety and amount of big data continue to grow. Much of that growth comes from unstructured data.

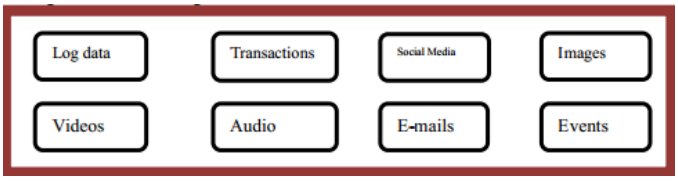

Fig1.1 Sources of Big data

III. KEY CHARACTERISTICS OF THE BIG DATA

Big Data starts with large-volume, heterogeneous, autonomous sources with distributed and decentralized control, and seeks to explore complex and evolving relationships among data. These characteristics make it an extreme challenge for discovering useful knowledge from the Big Data. The term Big Data literally concerns about data volumes, the key characteristics of the Big Data are

A. Huge with heterogeneous and diverse data sources: *One of the fundamental characteristics of the Big Data is the huge volume of data represented by heterogeneous and diverse dimensionalities. This huge volume of data comes from various sites like Twitter, Myspace, Orkut and LinkedIn etc.*

B. Decentralized control: *Autonomous data sources with distributed and decentralized controls are a main characteristic of Big Data applications. Being autonomous, each data source is able to generate and collect information without involving (or relying on) any centralized control. This is similar to the World Wide Web (WWW) setting where each web server provides a certain amount of information and each server is able to fully function without necessarily relying on other servers.*

C. Complex data and knowledge associations: *Multi structure, multisource data is complex data, Examples of complex data types are bills of materials, word processing documents, maps, time-series, images and video. Such combined characteristics suggest that Big Data require a "big mind" to consolidate data for maximum values.*

IV. THREE V'S IN BIG DATA

Doug Laney was the first one talking about 3V"s in Big Data Management

Volume: The amount of data. Perhaps the characteristic most associated with big data, volume refers to the mass quantities of data that organizations are trying to harness to improve decision-making across the enterprise. Data volumes continue to increase at an unprecedented rate..

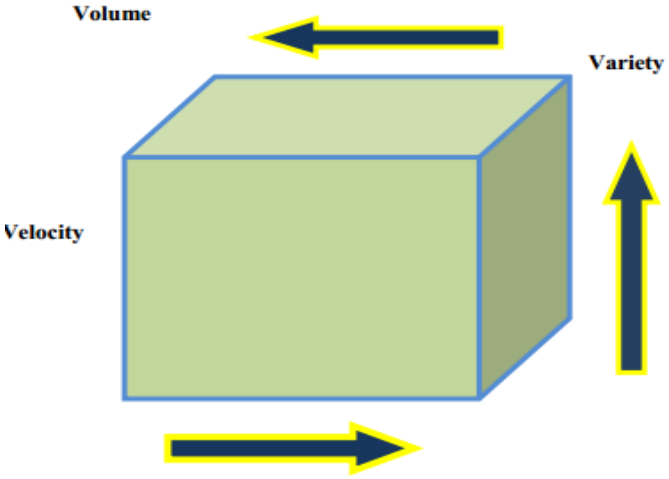

Fig1.2 Three V's in Big Data

Variety: Different types of data and data sources. Variety is about managing the complexity of multiple data types, including structured, semi-structured and unstructured data. Organizations need to integrate and analyze data from a complex array of both traditional and non-traditional information sources, from within and outside the enterprise. With the explosion of sensors, smart devices and social collaboration technologies, data is being generated in countless forms, including: text, web data, tweets, audio, video, log files and more.

Velocity: Data in motion. The speed at which data is created, processed and analyzed continues to accelerate.

Nowadays there are two more V"s

Variability: There are changes in the structure of the data and how users want to interpret that data.

Value: Business value that gives organization a compelling advantage, due to the ability of making decisions based in answering questions that were previously considered beyond reach..

V. HADOOP AND ITS COMPONENTS

Apache Hadoop is evolving as the best new approach to unstructured data analytics. Hadoop is an open source framework that uses a simple programming model to enable distributed processing of large

International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)
http://www.ijcsjournal.com Volume 3, Issue 1, No 5, 2015. ISSN: 2348-6600
Reference ID: IJCS-091 PAGE NO: 520-524.

data sets on clusters of computers.Hadoop offers several key advantages for big data analytics, including:

- *Store any data in its native format. Because data does not require translation to a specific schema, no information is lost.*
- *Scale for big data. Hadoop is already proven to scale by companies like Facebook and Yahoo!, which run enormous implementations.*
- *Deliver new insights. Big data analytics is uncovering hidden relationships that have been difficult, time consuming, and expensive—or even impossible—to address using traditional data mining approaches*

***HDFS**: The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.*

VI. MAP REDUCE

Map Reduce is a framework for processing parallelizable problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes are on the same local network and use similar hardware) or a grid (if the nodes are shared across geographically and administratively distributed systems, and use more heterogeneous hardware). Processing can occur on data stored either in a file system (unstructured) or in a database (structured). Map Reduce can take advantage of locality of data, processing it on or near the storage assets in order to reduce the distance over which it must be transmitted.

"Map" step: Each worker node applies the "map()" function to the local data, and writes the output to a temporary storage. A master node orchestrates that for redundant copies of input data, only one is processed.

"Shuffle" step: Worker nodes redistribute data based on the output keys (produced by the "map ()" function), such that all data belonging to one key is located on the same worker node.

"Reduce" step: Worker nodes now process each group of output data, per key, in parallel.

Map Reduce allows for distributed processing of the map and reduction operations. Provided that each mapping operation is independent of the others, all maps can be performed in parallel – though in practice this is limited by the number of independent data sources and/or the number of CPUs near each source. Similarly, a set of 'reducers' can perform the reduction phase, provided that all outputs of the map operation that share the same key are presented to the same reducer at the same time, or that the reduction function is associative. While this process can often appear inefficient compared to algorithms that are more sequential, Map Reduce can be applied to significantly larger datasets than "commodity" servers can handle – a large server farm can use Map Reduce to sort a petabyte of data in only a few hours. The parallelism also offers some possibility of recovering from partial failure of servers or storage during the operation: if one mapper or reducer fails, the work can be rescheduled – assuming the input data is still available.

Logical View

The Map and Reduce functions of Map Reduce are both defined with respect to data structured in (key, value) pairs. Map takes one pair of data with a type in one data domain, and returns a list of pairs in a different domain:

Map (K1, v1) → list (K2, v2)

The Map function is applied in parallel to every pair in the input dataset. This produces a list of pairs for each call. After that, the Map Reduce framework collects all pairs with the same key from all lists and groups them together, creating one group for each key.

The Reduce function is then applied in parallel to each group, which in turn produces a collection of values in the same domain:

Reduce (K2, list (v2)) → list (v3)

Each Reduce call typically produces either one value v3 or an empty return, though one call is allowed to return more than one value. The returns of all calls are collected as the desired result list.

Thus the Map Reduce framework transforms a list of (key, value) pairs into a list of values. These behaviors is different from the typical functional programming map and reduce combination, which accepts a list of arbitrary values and returns one single value that combines all the values returned by map.

Example 1: As another example, imagine that for a database of 1.1 billion people, one would like to compute the average number of social contacts a person has according to age. In SQL, such a query could be expressed as:

```
SELECT age, AVG(contacts)
FROMsocial.person
GROUPBY age
ORDERBY age
```

Using Map Reduce, the K1 key values could be the integers 1 through 1100, each representing a batch of 1 million records, the K2 key value could be a person's age in years, and this computation could be achieved using the following functions:

International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)

http://www.ijcsjournal.com Volume 3, Issue 1, No 5, 2015. ISSN: 2348-6600
Reference ID: IJCS-091 PAGE NO: 520-524.

function Map is
input:integer K1 between 1 and 1100, representing a batch of 1 million social.person records
for eachsocial.person record in the K1 batch do
let Y be the person's age
let N be the number of contacts the person has
produce one output record (Y,(N,1)) repeat
end function
function Reduce is
input: age (in years) Y
for each input record (Y,(N,C)) do
Accumulate in S the sum of N*C
Accumulate in Cnew the sum of C
repeat
let A be S/Cnew
produce one output record (Y,(A,Cnew))
end function
The Map Reduce System would line up the 1100 Map processors, and would provide each with its corresponding 1 million input records. The Map step would produce 1.1 billion(Y,(N,1)) records, with Y values ranging between, say, 8 and 103. The Reduce step would result in the much reduced set of only 96 output records (Y,A), which would be put in the final result file, sorted by Y.

The count info in the record is important if the processing is reduced more than one time. If we did not add the count of the records, the computed average would be wrong, for example:
-- Map output #1: age, quantity of contacts
10, 9
10, 9
10, 9
-- Map output #2: age, quantity of contacts
10, 9
10, 9
--- Map output #3: age, quantity of contacts
10, 10

If we reduce files #1 and #2, we will have a new file with an average of 9 contacts for a 10 year old person ((9+9+9+9+9)/5):
-- reduce step #1: age, average of contacts
10, 9

VII. PERFORMANCE AND CONSIDERATIONS

Map Reduce programs are not guaranteed to be fast. The main benefit of this programming model is to exploit the optimized shuffle operation of the platform, and only having to write the map and reduce parts of the program. Additional modules such as the Combiner function can help to reduce the amount of data written to disk, and transmitted over the network.For processes that complete fast, and where the data fits into main memory of a single machine or a small cluster, using a Map Reduce framework usually is not effective: since these frameworks are designed to recover from the loss of whole nodes during the computation, they write interim results to distributed storage. This crash recovery is expensive, and only pays off when the computation involves many computers and a long runtime of the computation - a task that completes in seconds can just be restarted in the case of an error; and the likelihood of at least one machine failing grows quickly with the cluster size. On such problems, implementations keeping all data in memory and simply restarting a computation on node failures, or - when the data is small enough - non-distributed solutions will often be faster than a Map Reduce system

VII. CONCLUSION AND FUTUTRE WORK

The analytics of enterprise data toward meaningful insights into the information that exists over a period of time in that context is why Big Data Analytics makes it different from traditional data warehouse analytics. The use of text analytics in real-world applications is growing very fast as organizations realize the untapped potential that is possible if textual data are analyzed and integrated in decision making.

An interesting and important goal of analyzing unstructured data such as customer complaints, issues, opinions or comments is to get a grasp on what they perceive about an entity. An entity can be a company's brand image, product, service, person, group or an organization. Are consumers' perceptions good, bad or neutral? What attributes (features) of the product or service they feel good or bad about? What do the customers think of the various attributes of a company's product such as quality, price, durability, safety, ease of use? Typically, if customer feels good towards an entity, it is classified as a positive sentiment. If the perception towards the entity is bad, it can be considered as negative sentiment. A third kind of perception in which customer has neither good nor bad opinion implies a neutral sentiment. Social media sites such as Twitter and Face book contains enormous volumes of customer opinions and comments on virtually all major organizations, events and products. This creates an unprecedented opportunity to mine text data in real-time to and analyzes sentiment trends fluctuations over a period of time. Accordingly, this research has provided the sentiment of the people with examples of the various big data tools and technologies which can be applied. This gives users an idea of the necessary technologies required, as well as developers an idea of what they can do to provide more enhanced solutions for big data analytics in support of decision making. Thus, the support of sentimental analysis with big data analytics to decision making was depicted.

International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)

http://www.ijcsjournal.com Volume 3, Issue 1, No 5, 2015. ISSN: 2348-6600
Reference ID: IJCS-091 PAGE NO: 520-524.

VIII. REFERENCES

Berry, M.andLinoff, G., Master Data Mining: The Art and Science of Customer Relationship Management, Wiley publisher, 2000.

2. Fayadd, U., Piatesky -Shapiro, G., and Smyth, P, From Data Mining To Knowledge Discovery in Databases", The MIT Press, ISBN 0–26256097–6, Fayap, 1996.

3. Gorunescu, F, Data Mining: Concepts, Models, and Techniques, Springer, 2011.

4. Han, J., and Kamber, M. ,Data mining: Concepts and techniques, Morgan-Kaufman Series of Data Management Systems San Diego:Academic Press, 2001.

5. Heikki, Mannila, Data mining: machine learning, statistics and databases, IEEE, 1996.

6. Jing He,Advances in Data Mining: History and Future, Third international Symposium on Information Technology Application, 978-0-7695- 3859-4, IEEE, 2009.

7. Kubick, W.R.: Big Data, Information and Meaning. In: Clinical Trial Insights, pp. 26–28 (2012)

8. NeelamadhabPadhy, Dr. Pragnyaban Mishra and RasmitaPanigrahi, "The Survey of Data Mining Applications and Feature Scope, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)", vol.2, no.3, June

9. Piatetsky-Shapiro, Gregory, The Data-Mining Industry Coming of Age,"IEEE Intelligent Systems, 2000.

10. Russom, P.: Big Data Analytics. In: TDWI Best Practices Report, pp. 1–40 (2011).