EMPOWERING CHATBOTS WITH BUSINESS INTELLIGENCE BY BIG DATA INTEGRATION

Article · February 2018

DOI: 10.26483/ijarcs.v9i1.5398

Reshmi Sankar
Cochin University of Science and Technology

DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5398

ISSN No. 0976-5697

Volume 9, No. 1, January-February 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

EMPOWERING CHATBOTS WITH BUSINESS INTELLIGENCE BY BIG DATA INTEGRATION

Reshmi.S
Department of Computer Applications
Cochin University of Science and Technology
Cochin, India

Kannan Balakrishnan
Department of Computer Applications
Cochin University of Science and Technology
Cochin, India

Abstract: Chatbots are one of the most widely used technologies to implement virtual assistance. Presently, chatbot based virtual assistants are being used by many web administrators to mediate access to data and to carry out generic conversations with the users. Such virtual assistants are getting a lot of attention from the business organizations, as it can help in improving customer care support; reduce the costs in customer service centers and can handle multiple clients at a time.

Big data analytics is the process of collecting, organizing and analysing large data sets to discover patterns and unknown correlations hidden in the data, such as usage statistics and customer preferences, which can serve as valuable business information. This paper describes the implementation of a chatbot framework with an interface to big data. This implementation would provide mass knowledge analysis capability to chatbots from distributed environments, which can further the spectrum of usage of such intelligent agents.

Keywords: Chatbot, Knowledge base, Virtual assistant, AIML, Big Data, Hadoop, Hive

I. INTRODUCTION

An intelligent virtual assistant is a software agent which employs the possibilities of Artificial Intelligence, in order to perform tasks or services for a user, based on his/her inputs. Virtual assistants are being increasingly used by organizations to provide their users with better customer experience and to manage dialogues on issues relating to the activities of the organization or its offerings. Some of the common tasks performed by virtual assistants include, responding to the user's queries, acting as a guide or tutor, taking customers on a tour of the website, guiding them in their shopping decisions etc. A virtual assistant does not only answer questions - it also tries to hold a conversation, mimicking human behaviour, so that the users would feel that they are interacting with a real human. Such an assistant may basically consist of a dialog system, an avatar, as well an expert system to process queries efficiently.

One of the familiar technologies to implement virtual assistance is chatbot. Chatbots, also known as conversional agents, are software frameworks that can respond to natural language inputs and attempts to hold a conversation in a way that imitates a real person. Chatbots communicate with their human partners through various frameworks ranging from a simple text interface to speech recognition features.

Big data, the vast amount of data generated due to digitalization, has a tremendous impact on business processes. According to business research studies, big data might help companies to make better strategic decisions, understand customer needs, efficiently control processes and reduce costs [1]. Big data analytics enables the analysis of a combination of structured, semi-structured and unstructured data. Such an analysis helps to understand the information contained in the data in a better way and requires the use of specialised software frameworks like Hadoop, Spark etc.

II. OVERVIEW OF CHATBOTS

One of the main aims of chatbots has always been to imitate and resemble humans as far as possible in order to hide their artificial nature while interacting with the users. Many efforts have been devoted in the field of development of chatbots to interact with the user in natural way, using various architectures and features, broadening their scope and usage widely throughout the history. Pattern matching, finite-state-machines and frame-based models are the main methodologies of conversional agent design [2]. Many research works have also been focused in the development of chatbots to enable them to interact with the users in natural language in a seamless way. Many areas like artificial tutoring, automated customer service, health industry, e-commerce, finance etc., which require the human-computer interactions, has been been taken over by chatbots.

Main components of a chatbot include the knowledge base, an interpreter program and chat engine [3]. Knowledge base encapsulates the intelligence of the system, and generally composed of keywords/phrases and responses associated to each keyword/phrase. Common implementation of the knowledge base involves the use of dat files or text files, databases, script files and XML files. The interpreter program consists of an analyser and generator for communicating with the user interface. Analyser reads input statement from the user and analyses syntax and semantics of the sentence. Analyser acts as a pre-processor and uses different normalization techniques like pattern fitting, substitution and sentence splitting etc., for the easy analysis of the user input. The chat engine tries to match the pre-processed output of the analyser to identify suitable response(s) using pattern matching algorithms, with the help of the knowledge base. Generator processes the response given by the chat engine and generates appropriate grammatically correct sentences to be displayed to the user.

The early classic chatbot, named Eliza [4] simulated a Rogerian psychotherapist by Joseph Weizenbaum. It was inspired by the ideas of Turing [5], who argued that it was possible to build machines capable of acting like humans. This bot worked based on pattern matching algorithms and sentence reconstruction, without in-depth knowledge or processing of natural language. The program proved to be amazingly efficient in sustaining people's attention during the conversation and the success of the original program has influenced the development of many other bots.

Parry chatbot, developed by Colby, simulates a patient and has been intended as a study of the nature of paranoia and is capable of expressing beliefs, fears, and anxieties [6]. Another worth mentioning chatbot is Jabberwacky, with the ability to learn new responses based on user interactions. This feature of learning by interaction rather than being driven from a static database made Jabberwacky distinct from other contemporary chatbots [7]. In order to achieve this of Jabberwacky keeps track of every user response and finds the most appropriate response using contextual pattern matching techniques [8].

A.L.I.C.E (Artificial Linguistic Internet Computer Entity) is a natural language processing chatbot [9], developed by Dr. Richard Wallace. It has its own markup language called AIML (Artificial Intelligence Markup Language), and earned the Loebner Prize in 2000 and 2001. ALICE applies heuristic pattern matching algorithm to inputs to obtain suitable matching pattern in AIML and this algorithm uses a depth first search technique with backtracking. The knowledge base of this system is composed of AIML files, which is an extension of the widely used XML format. This XML compliant dialect for encoding the behavior of a chatbot in a standardized form can be exchanged between different chatbot interpreters and implementations. AIML is highly recursive and typically a single input-response pattern will have many alternative patterns matches. An AIML dataset typically consists of pattern and template that can be matched to the user's input statement and the corresponding response.

Chatbots have also been stepped into the health industry and can provide many health-related services like answering medical queries which allows patients to receive immediate treatment information without waiting doctors for long hours. Incidentally, early chatbots Eliza and Parry, were designed to chat with patients. Another notable chatbot implementation in the health domain is the Pharmabot [10], which can act as consultant pharmacist that will give appropriate and safe medication of generic drugs for children based on the information collected from the user through chatting. Pharmabot can act as a medicine consultant for the patients or parents who are confused with the generic medicines and assist the user in taking the right generic medicine for a certain ailment.

III. BIG DATA ANALYTICS

Big data is a term for data sets that are so large or complex, and hence traditional techniques of data processing may turn out to be inadequate. It is generated on a very large scale from social media websites, multimedia sources and other forms of network related data along with real time data generated from sensors and devices. Big data analytics is the process of examining this large amount of data, to uncover hidden patterns, unknown correlations etc., which may correspond to market trends, customer preferences and other useful information, depending on the origin of the data. The analytical findings can lead to more effective use of the information gained, and can be applied in various fields like marketing, sales, customer care etc; to improve operational efficiency, to gain competitive advantages and other business benefits.

The analysis of big data sets in real-time requires a platform like Hadoop to store large data sets across a distributed cluster and process these data from multiple sources. Hadoop is an open source software framework for storing big data and comprises of a set of tools for processing very large data sets in a distributed computing environment [11]. It provides massive storage facility for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs [12]. However, Hadoop is not a conventional type of database, rather a software ecosystem that allows for massive parallel computing.

The core modules of Hadoop comprise of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce, along with a resource management platform called YARN [12]. HDFS is a distributed file system that stores data across multiple machines without prior organization. MapReduce is a programming model for large scale data processing in parallel, which basically takes intensive data processes and spreads the computation across a potentially endless number of servers. YARN (Yet Another Resource Negotiator) is the resource management platform for scheduling and managing resource requests from distributed applications.

There are also supplementary tools like Pig, Hive, Sqoop, Avro etc., which have been developed on top of Hadoop framework, which can act as data access framework. One of these tools, Hive, is a data warehouse software which facilitates easy data management of large datasets residing in distributed storage and provides data summarization and ad hoc querying. Hive employs a declarative SQLish language called HQL (Hive Query Language) which is very similar to SQL, for querying the data. The queries in HQL are translated into MapReduce programs by Hive and are executed in runtime [13]. Fig. 1. illustrates the Hadoop framework and its associated tools [14].

HCatalog is a table and storage management layer for Hadoop and to holds the metadata and location of the data in a Hadoop cluster [15]. This allows scripts and MapReduce jobs to be decoupled from data location and metadata, enabling different data processing tools to read and write data effortlessly on Grid.

The Hadoop-MapReduce framework has rapidly become popular for big data processing. Scalability and fail over properties of Hadoop-MapReduce are the main reasons attributed to such popularity [16]. The framework is also known for its ease of use which allows even non-expert users to easily run analytical tasks over big data.

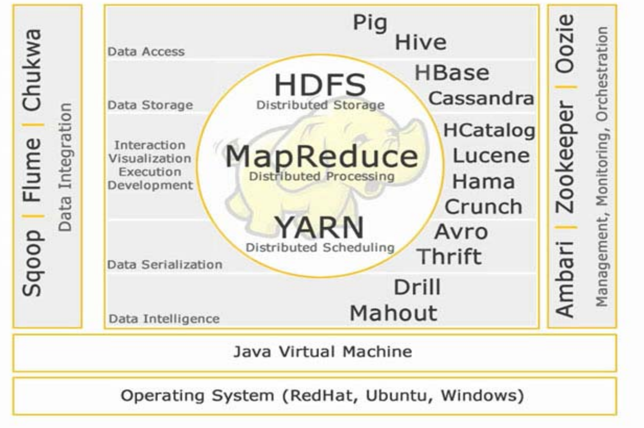


Figure 1. Hadoop framework and tools.

IV. PROPOSED SYSTEM

Chatbots can respond to a set of predefined queries which are generally stored in a knowledge base. The ability of the chatbots to respond to the variants of the same query makes it more efficient and attractive and this ability, to a large extend, is determined by the algorithms used for pattern matching. Another important factor is the amount of information made available to the chatbot, using which it can frame the response to the queries and is determined by the implementation of the knowledgebase. As the information available to the chatbot increases, the number of queries to which the chatbot can responds to also increases, making it more efficient and useable.

Generally, the knowledge base is implemented using text files, AIML files or using relational databases. Even though text files are one of the easiest methods to implement the knowledge base, it does not provide any inbuilt mechanism for easy pattern matching. With relational databases, one can store data in a structured and organised way. However, it may not be a suitable option for storing generic query-response patterns. Thus, the use of AIML for implementing the knowledge base has gained focus. Though, AIML is very easy to use and learn, it is difficult to incorporate dynamic changes and can only store limited amount of data, in a simple category-pattern format.

This paper discusses the integration of big data as knowledge base into the chatbot framework along with a modified AIML knowledge base. This hybrid knowledge base model allows chatbot to connect to the big data environment, there by imparting knowledge analysing capability from large amount of data, from distributed environment. The proposed system is built on the famous ALICE [9] chatbot framework, which is based on the AIML technology.

V. METHODOLOGY

In the existing ALICE framework, the chat engine identifies a suitable response for the user's query using pattern matching algorithms with the help of the knowledge base. ALICE engine uses AIML files as the knowledge base which stores set of predefined queries and its variants. In the proposed hybrid knowledge base model, the response for the user's query can come either from the AIML or from the big data knowledge base. While the responses which are more permanent in nature can be stored in the AIML, those responses which need some sort of analysis or which are dynamic in nature can be fetched from the big data.

To achieve this, an additional knowledge base engine (KB engine) has been proposed along with the existing system, which can interface with the big data framework for fetching factual data for responding to certain queries. Since the incoming query is first searched in the AIML, a suitable mechanism is needed in the AIML to instruct the chat engine, to redirect the query to the KB engine for searching in the big data. Hence modified AIML constructs have been implemented to incorporate such KB engine processing. The KB engine communicates with the big data knowledge base through hive interface and finds proper response from the big data using modified AIML responses redirected by the chat engine. Fig. 2 illustrates the architecture of the proposed system with the main elements involved in it.

In order to setup a prototype environment, a single node Hadoop data cluster on Windows machine is configured. Typically, Hadoop framework runs on UNIX/LINUX computers, however single node Hadoop cluster can be setup using a virtual machine installed on Windows machine. This can be achieved using Hortonworks Sandbox installed along with Oracle Virtual Box. The Hortonworks Sandbox packages a single node implementation of enterprise-ready open source Apache Hadoop distribution based on a centralized architecture (YARN) along with a virtual environment that can run in the cloud or on personal machine [17]. The Sandbox is a straightforward, pre-configured, learning environment to make evaluation and experimentation fast and easy.

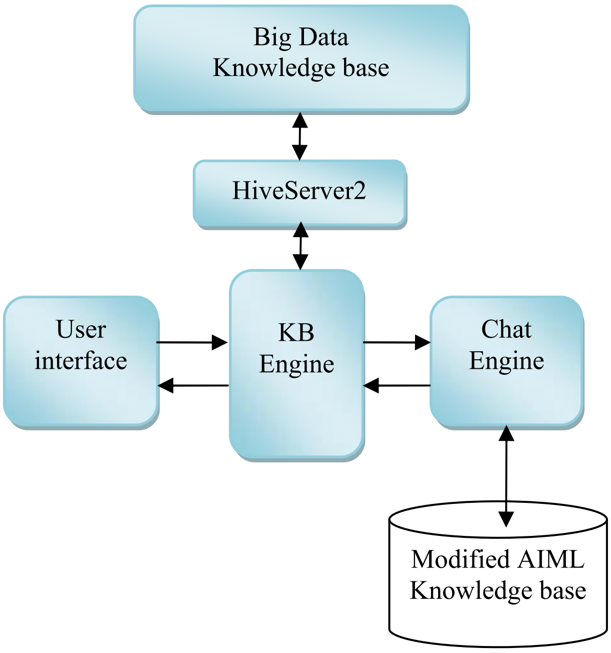


Figure 2. Proposed system architecture.

For sending queries to this big data environment and to retrieve data, chatbot framework has to establish a connection with the environment. HiveServer2 (HS2) is a server interface that enables remote clients to execute queries against Hive and retrieve the results. HS2 supports multi-client concurrency and is designed to provide better support for open API clients like JDBC and ODBC [18]. The chatbot communicates with HS2, over a driver connection using JDBC, which intern connects with Hive. Hive facilitates easy data management of large datasets residing in distributed storage. It provides a SQL-like interface to create hive query to retrieve data from Hadoop [19].

In the chatbot framework, the connectivity to the Hive to access Hadoop data has been implemented through the KB engine. Thus, the KB engine is designed to integrate the big data as a knowledge base and act as an interpreter between big data and chat engine. The Hive connectivity API driver enables to access Hadoop data through KB engine and sent HQL statements to Hive to fetch information from big data knowledge base. The users query is intercepted by the KB engine and is passed on to the chat engine for processing. The chat engine processes the users query by parsing it and comparing it against the AIML templates. If the AIML, readily contains a response, it is passed directly to the user, through the KB engine. In case if the response is dynamic in nature and need to be fetched from big data knowledge base, then it is represented using modified AIML template whose format is as shown in the Fig. 3.

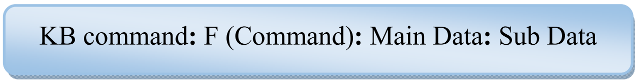

KB command**:** F (Command)**:** Main Data**:** Sub Data

Figure 3. KB Engine command structure.

The template starts with a token “KB” which indicates that it should be processed by the KB engine. It is followed by a command that can be mapped with the action which should be executed by the KB engine. F (command) indicates, function to act by KB engine. The template also stores the main and sub data which are required for processing the action. This template is provided to the KB engine for evaluation which will in turn lead to the execution of appropriate command action.

The generalized command response to fetch data from big data knowledge base has the following format,

KB engine command: ReadBigdataKB: HQL query: Number of Fields to process.

Process flow of proposed system is depicted as sequence diagram in Fig. 4.

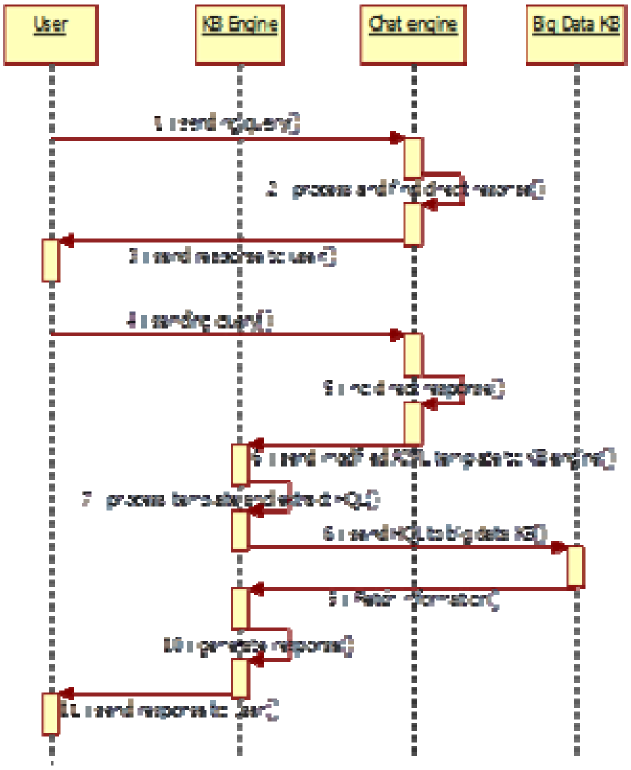
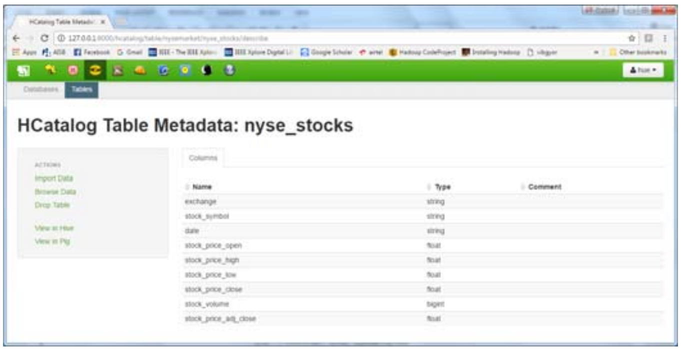

User
KB Engine
Chat engine
Big Data KB
1 : send a query()
2 : process and find direct response()
3 : send response to user()
4 : sending query()
5 : no direct response()
6 : send modified AIML template to KB engine()
7 : process template and create HQL()
8 : send HQL to big data KB()
9 : fetch information()
10 : generate response()
11 : send response to user()

Figure 4. Sequence diagram of proposed system.

VI. RESULTS AND DISCUSSIONS

Modified the AIML knowledge base along with chat engine and integrated big data framework as knowledge base which empowers business intelligence to the chatbot. By integrating big data to chatbot, even non-expert users without the knowledge of big data framework and HQL queries can fetch information through simple chats from a large volume of unstructured data and distributed ecosystem which is not easily accessed by traditional chatbot applications.

For the demonstration of this implementation the stock ticker data from the New York Stock Exchange from the years 2000-2001 [20] has been made use of. The data has been uploaded to HDFS and registered with HCatalog utility to generate table structure, so that the location and metadata can be accessed in Hive. Fig.5. shows the metadata of the uploaded New York Stock Exchange data through HCatalog.

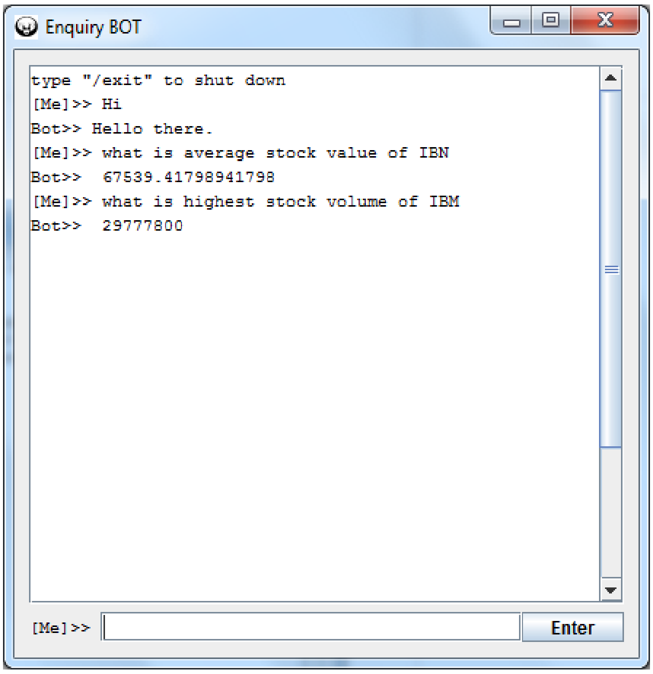


HCatalog Table Metadata: nyse_stocks

Name	Type	Comment
exchange	string	
stock_symbol	string	
date	string	
stock_price_open	float	
stock_price_high	float	
stock_price_low	float	
stock_price_close	float	
stock_volume	bigint	
stock_price_adj_close	float	

Figure 5. Metadata of uploaded data through HCatalog.

Assume that a stock market user needs to know the average stock volume of company, with existing architecture cannot meet this query as the AIML framework supports only static query. The proposed architecture solves this limitation, by enabling such dynamic responses to be generated through the KB engine by analyzing the big data. User interface of a simple interaction session between a user and the proposed chatbot is shown in Fig. 6.



```
type "/exit" to shut down
[Me]>> Hi
Bot>> Hello there.
[Me]>> what is average stock value of IBN
Bot>>  67539.41798941798
[Me]>> what is highest stock volume of IBM
Bot>>  29777800
```

Figure 6. User interface of Big Data integrated Chatbot.

VII. CONCLUSION

Chatbots are becoming popular in domains where human computer interaction takes place, like virtual assistance, artificial tutoring, e-commerce, health care, finance etc. The integration of big data as knowledge base into the chatbots can enable the generation of dynamic responses to user queries and improve the analytical capability of chatbots with data from distributed environment. This enabling technology directly opens up the world of big data to chatbots, allowing the chatbots to be used as a business intelligence analytics tool as well.

VIII. ACKNOWLEDGMENT

The authors gratefully acknowledge the Department of Computer Applications, Cochin University of Science and

Technology, for extending all the facilities for carrying out this work.

IX. REFERENCES

[1] J. Heidrich, A. Trendowicz, and C. Ebert, "Exploiting Big Data's Benefits," IEEE Softw., vol. 33, no. 4, pp. 111–116, 2016.

[2] A. Augello, G. Pilato, G. Vassallo, and S. Gaglio, "A Semantic Layer on Semi-Structured Data Sources for Intuitive Chatbots," in 2009 International Conference on Complex, Intelligent and Software Intensive Systems, 2009, pp. 760–765.

[3] B. Hettige and A. S. Karunananda, "First Sinhala Chatbot in action," in Proceedings of the 3rd Annual Sessions of Sri Lanka Association for Artificial Intelligence (SLAAI), 2006, no. September, pp. 4–10.

[4] J. Weizenbaum, "Eliza - A Computer Program For the Study of Natural Language Communication between Man and Machine," Commun. ACM, vol. 9, no. 1, pp. 36–45, 1966.

[5] A. M. Turing, "Computing machinery and intelligence," Mind, New Ser., vol. 59, no. 236, pp. 433–460, 1950.

[6] G. Güzeldere and S. Franchi, "Dialogues with colorful personalities of early AI," in Constructions of the Mind. Artificial Intelligence and the Humanities. Special Issue of the Stanford Humanities Review, vol. 4, no. 2, 1995, pp. 161–170.

[7] R. Carpenter and J. Freeman, "Computing machinery and the individual: the personal Turing test," 2005.

[8] R. Carpenter, "Jabberwacky chatbot." [Online]. Available: http://www.jabberwacky.com/j2about. [Accessed: 05-Dec-2016].

[9] R. S. Wallace, "ALICE." [Online]. Available: http://www.alicebot.org/about.html. [Accessed: 05-Dec-2016].

[10] B. E. V Comendador, B. M. B. Francisco, J. S. Medenilla, S. M. T, and T. B. E. Serac, "Pharmabot□: A Pediatric Generic Medicine Consultant Chatbot," J. Autom. Control Eng., vol. 3, no. 2, pp. 137–140, 2015.

[11] "Apache Hadoop." [Online]. Available: http://hadoop.apache.org/. [Accessed: 07-Oct-2016].

[12] S. K. Garima Rani, "Hadoop Technology to Analyze Big Data," Int. J. Eng. Dev. Res., vol. 3, no. 4, pp. 949–952, 2015.

[13] "Apache Hive documentation." [Online]. Available: https://cwiki.apache.org/confluence/display/Hive/Home. [Accessed: 20-Nov-2016].

[14] "Hadoop Ecosystem and its components." [Online]. Available: http://www.edupristine.com/blog/hadoop-ecosystem-and-components. [Accessed: 17-Oct-2016].

[15] "HCatalog." [Online]. Available: https://cwiki.apache.org/confluence/display/Hive/HCatalog+UsingHCat. [Accessed: 11-Dec-2016].

[16] J. Dittrich and J. Quian, "Efficient Big Data Processing in Hadoop MapReduce," Proc. VLDB Endow., vol. 5, no. 12, pp. 2014–2015, 2012.

[17] "Learning the Ropes of the Hortonworks Sandbox." [Online]. Available: http://hortonworks.com/hadoop-tutorial/learning-the-ropes-of-the-hortonworks-sandbox/. [Accessed: 11-Dec-2016].

[18] "HiveServer2 Clients." [Online]. Available: https://cwiki.apache.org/confluence/display/Hive/HiveServer2+Clients#HiveServer2Clients-JDBC. [Accessed: 11-Dec-2016].

[19] "Process data with Apache Hive." [Online]. Available: http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-hive/. [Accessed: 11-Dec-2016].

[20] "New York Stock Exchange data." [Online]. Available: https://s3.amazonaws.com/hw-sandbox/tutorial1/NYSE-2000-2001.tsv.gz. [Accessed: 20-Oct-2016].