Social Media Based NPL System to Find and Retrieve ARM Data: Concept Paper

Ranjeet Devarakonda (*Senior Member, IEEE)*, Michael Giansiracusa, Jitendra Kumar, and Harold Shanafield III

ARM Data Center, Oak Ridge National Laboratory, Oak Ridge, TN 37831

***Abstract*— Information connectivity and retrieval has a role in our daily lives. The most pervasive source of online information is databases. The amount of data is growing at rapid rate and database technology is improving and having a profound effect. Almost all online applications are storing and retrieving information from databases. One challenge in supplying the public with wider access to informational databases is the need for knowledge of database languages like Structured Query Language (SQL). Although the SQL language has been published in many forms, not everybody is able to write SQL queries. Another challenge is that it may not be practical to make the public aware of the structure of the database. There is a need for novice users to query relational databases using their natural language. To solve this problem, many natural language interfaces to structured databases have been developed. The goal is to provide more intuitive method for generating database queries and delivering responses. Social media makes it possible to interact with a wide section of the population. Through this medium, and with the help of Natural Language Processing (NLP) we can make the data of the Atmospheric Radiation Measurement Data Center (ADC) more accessible to the public. We propose an architecture for using Apache Lucene/Solr [1], OpenML [2,3], and Kafka [4] to generate an automated query/response system with inputs from Twitter[5], our Cassandra DB, and our log database. Using the Twitter API and NLP we can give the public the ability to ask questions of our database and get automated responses.**

***Index Terms*— machine learning, natural language processing, social media interaction, stream pipelining**

I. INTRODUCTION

ADC currently use Apache Lucene based Solr indexing to achieve fast search responses of our ZFS file structure. ARM's data discovery (https://www.archive.arm.gov/discovery) tool directly queries the Solr index and displays the search results on the user-interface. Currently, ARM distributes about 22 million data files via the data discovery tool. The Solr instance uses Lucene's inverted index. This inverted indexing system allows us to search a smaller subset of the data when executing queries, the results of which direct our system to the appropriate files. Solr is built on a high-performance text search engine library written in Java [1]. We propose to use OpenML in conjunction with Apache Kafka to create an automated workflow that processes datastreams, queries our database, and generates automatic responses. One of the applications of the proposed system would be, ARM users could tweet a specific science question—that are tracked by our Kafka Zookeeper service [2] automatically, from the twitterapi—convert questions into ARM's terminology, using NLP scripts—query Solr index and other ARM databases, for answers—send response back to the user.

OpenML is an online machine learning platform that leverages community involvement to deliver a wide variety of well-reviewed machine learning algorithms [3,4]. We will use algorithms adapted from OpenML to process data from a program designed to monitor twitter for a regular expression or phrase which we can answer using ADC data. Other inputs will be data streams from our Cassandra database and streams of log files generated for reporting and monitoring purposes. These streams of data will be consumed and consolidated into Apache Kafka, which is a real-time streaming data pipeline manager. Kafka is run on clustered environments with fault-tolerant storage. Kafka also has implementation for database connectivity, stream processors, ingesting producer data streams, and output data streams to consumers [5].

DOI: 10.1109/BigData.2017.8258525

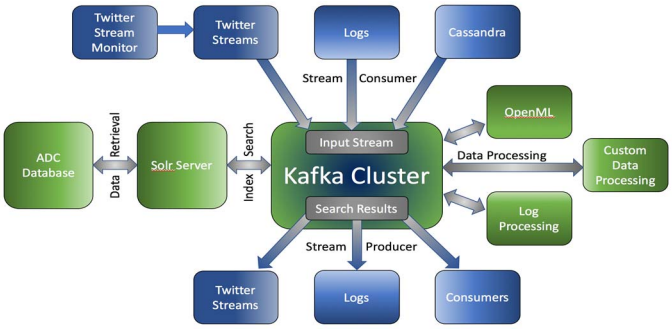

Figure 1. Data producer and consumer workflow using Kafka

We will use Apache Kafka to manage the processing pipeline which will allow us to deploy scalable machine learning across several systems and leveraging multiple monitoring applications. Kafka provides the connectivity that allows multiple unique applications to use data from a wide variety of apps and databases. Feature data is ingested into Kafka from the various apps and databases that host it. The data can then be used to train and build models which can range from data warehouses, big data environments such as Cassandra and Spark, or a server running python or scala scripts. Models built using this framework can be published in production where apps that use the same model parameters can apply it to incoming data. This design allows us to create one system which can be used to supply, build, apply, and monitor analytic models. Using Kafka as a scalable, distributed message broker we will turn streaming twitter data, logs and records from Cassandra into answers delivered via social media, direct communication, or other means. This system will lower production time, increase availability of data, and facilitate deployment and monitoring by managing message flow in a central location. Kafka's stream producer will automate input and output. The workflow is visualized in figure 1. To increase visibility of data, machine learning will be used to build a social media bot which will answer natural language science questions using ARM data. The application will run as a Kafka stream and use the Twitter api, natural language processing modules, response processing, and twitter feedback monitoring. In standard English, tokenization is a trivial task; Sentences can be broken up into word tokens using spaces. Tweets pose a challenge for standard tokenizers due, in part, to the 140-character limit (soon to be 280 characters). Users often create abbreviations, acronyms, conjunctions, or use symbols to represent complex concepts. Machine learning is important in making connections between these creative representations of words and concepts to accurately derive meaning from 140 characters. Machine is used for part of speech tagging, theme extraction and sentiment processing as part of a comprehensive approach to natural language processing applied to social media [6]. These techniques are applied to inputs from social media to understand the questions asked by users and to gain an understanding about the effectiveness of our outputs by monitoring the responses to outputs. When a user question is answered, the responses to that output are tracked and undergo semantic analysis to determine the effectiveness of the system. This second level feedback will allow the system to generate better responses to questions as it learns how users respond to the outputs.

ACKNOWLEDGEMENT

Oak Ridge National Laboratory is managed by the UT-Battelle, LLC, for the U.S. Department of Energy under contract DEAC05-00OR22725.

REFERENCES

[1] Khilosiya, C., & Channe, H. P. (2014). Performance evaluation of searching using various indexing techniques in One of the applications of the proposed system would be, ARM users could tweet a science question—that are tracked by Kafka Zookeeper service automatically from the twitterapi—convert questions into ARM's terminology, using NPL scripts—query Solr index and other ARM databases for answers—send response back to the user automatically.
Lucene with Relational Databases. International Journal of Advanced Computer Research, 4(1), 212.

[2] https://kafka.apache.org/documentation/

[3] Vanschoren, J., Van Rijn, J. N., Bischl, B., & Torgo, L. (2014). OpenML: networked science in machine learning. ACM SIGKDD Explorations Newsletter, 15(2), 49-60.

[4] Van Rijn, J. N., Bischl, B., Torgo, L., Gao, B., Umaashankar, V., Fischer, S., ... & Vanschoren, J. (2013, September). OpenML: A collaborative science platform. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 645-649). Springer, Berlin, Heidelberg.

[5] Bhattacharya, D., & Mitra, M. (2013). Analytics on big fast data using real time stream data processing architecture (pp. 1-34). EMC Corporation.

[6] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., ... & Smith, N. A. (2011, June). Part-of-speech tagging for twitter: Annotation, features, and experiments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2 (pp. 42-47). Association for Computational Linguistics.