IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ANALYZING RESPONSES ON ONLINE DEVELOPER COMMUNITY: A CASE STUDY

[1]Kommu Gangadhara Rao

[1]Assistant Professor
[1]Department of Information Technology
[1]Chaitanya Bharati Institute of Technology, Hyderabad, India

Abstract: In recent years Stack Overflow has been criticized by various users on and off social media for being unwelcoming to new users. In fact, Stack Overflow also recognizes that this is a serious problem since they are losing old contributors and the criticism and widespread negative attitudes on the website. Because the website makes its data available online, we were able to form hypotheses and analyze the issue. We suspect that the hostile attitudes towards new users could be correlated with the maturity of scripting and programming languages or frameworks. Our work analyzes the developer community of Stack Overflow through the lens of users and languages heterogeneities. Students and young professionals would find the results useful when they decide which programming language to learn and how to get involved in the community wisely. Stack Overflow administors could adopt our algorithms to build the real-time dashboard to track the trends of the languages and provide data-driven insights for the developers

Index Terms **– Stack Overflow, Exploratory Analysis, Scripting Languages**

I. INTRODUCTION

The first section discusses the background of the project and the research hypothesis. The second section documents our data set and methods for preprocessing. The third section discusses our exploratory analysis. The fourth section details the main analysis which provides an answer to our research hypothesis. The final section provides a high level summary of our project. The extrapolated runtimes and discussion of challenges are included in each section or task where applicable.

1.1 Hypothesis

We question if answer providers on Stack Overflow become more impatient and meaner as scripting and programming languages or frameworks become more mature and popular. However, we need a reliable measurement of "impatience" or "hostile attitudes." Thus, we refined our question as follows: as programming languages mature, have sentiments of the answers become more negative? We hypothesize that as scripting and programming languages or frameworks become more mature and popular, the sentiments of Stack Overflow answer providers become more negative.

1.2 Testing

1. To test our hypothesis, we employ the following procedures:
2. We processed the data set so that it is in the form that is appropriate for our method of analysis.
3. We did an exploratory analysis on the top 15 scripting and programming languages or Frameworks based on their popularity
4. We then cross-checked the list with the scripting and programming languages or frameworks that are relevant to the class.
5. We conducted other relevant exploratory analysis to gain more understanding about users, Questions, and answer providers.
6. We conducted the main analysis.

1.3 Data Description

The data set was originally downloaded in bundle from Archive.org.[1]The table below details the breakdown of our data set. The data set comprises of 8 sub files. We only utilized the only four from them

Sl.No	File Name	Type	Size	Num of Lines	Csv File Size
1	Tags	XML	4.7 GB	54,467	5.6 GB
2	Users	XML	3.1 GB	10,097,980	1.97 GB
3	Badges	XML	3.4 GB	30,347,227	3.28 GB
4	Posts	XML	66 GB	43,872,994	22.5 GB

Table 1.1: Describing size of the Data

1.4 Data Preparation

Most of the cleaning and data processing was done using MPI in Python. While many of the analyses relied on combining and condensing data from Stack Overflow into a new dataset of different measurements, for which MRjob was well suited for, this was not the case for the file processing and cleaning. In these instances, each line of a file needed to be processed, but as each line could be processed independently of each other, this made MPI uniquely suited to handle these situations. As such, we used MPI to handle two main tasks, converting the raw data from the Stack Overflow data dump into a format that could be more easily used in the analyses, and for converting the output files from these analyses into a format that could be more easily used for summarizing our results

The raw data from the Stack Overflow data dump came in XML format, which was difficult to work within its initial form. XML files are hard to interpret on their own, as columns cannot be separated just by splitting each line by one character, like a CSV. Therefore, we wanted to use a pre-built parser to get the text data from the XML file and write it to a CSV file. For the XML parsing, we used xml.sax, as it could process files line-by-line and allowed for easy access to the keys and their values for each XML tag. As the values were not in the same order for each tag, knowing what key it corresponded to was important to putting that value in the right CSV column.

II. Exploratory Analysis

2.1 Questions: Who are the most active users?

The objective of this analysis is to gain more understanding about user activities. To carry out this task, we utilize the Spark framework which we learned in class. Since the Posts.csv file contains data on questions and answers that were posted by each user ID. We were able to do a simple counting of how many times a unique ID appears in the file and extrapolate the user activities from that. We define user activities as the number of questions and answers posted by a user.

Instead of using Spark's resilient distributed dataset (RDD), we opted for a Spark's dataframe to gain more exposure to the data structures provided by Spark. The Spark's dataframe is built on top of RDD, but the advantage of a dataframe is that it allows users to interact with it as a table with rows and columns. This proves to be very intuitive for us since we frequently use Pandas or R dataframe in other classes. The challenge of this task lies not in the computation complexity but in learning the syntax and figure out how to submit Spark jobs on Google dataproc. Learning how to launch a Spark cluster was not as difficult as we thought. In fact, it was fairly easy since we previously learned how to launch and submit a MapReduce(Hadoop) job on a dataproc cluster in a lab. This exercise proves to be valuable because we were able to apply skills we learned in class to a slightly different problem.

The runtime of this analysis with 1 master node and 3 workers is approximately 2 minutes 45 seconds which is very fast compared to other MRJob approaches that we did for other tasks. We wrote the code for the same analysis which yields slightly different output (not in terms of discrepancy) using a MapReduce framework. Running the task on a default Google dataproc setting takes about less than an hour. However, we were not able to compare the gain in performance from using Spark to MapReduce/MRJob unless we control for the number of nodes and the location of the server. A more rigorous comparison is definitely needed. Big data frameworks are certainly useful for this analysis because the file is 22.5 GB in size which cannot even be read on to memory or a Pandas dataframe.

The objective of this task is to determine how has the most popular questions asked on Stack Overflow changed over time. This helps us gauge how the developer community has changed over time and what are questions are most likely to be answered.

2.2 Where *do the users that answer the most come from?*

In this task, we try to get a sense for the locations of the most active answerers. This information will be helpful to determine if most active answerers are predominantly from English speaking nations or not. Depending upon this, it would strengthen or weaken findings of the main sentiment analysis of responses. If the population of answerers are mostly non-native English speakers, there choice of words and expression may differ considerably.

In order to find an answer to this question, information from two data files, Badges.csv and User.csv, is used. Users file includes information about all users and details about their account. A badge is a commendation given by Stack Overflow to users for achieving various milestones. For most categories, there are three different classes (gold, silver and bronze) and each of them has a unique badge name. Since we are interested in top answerers, we will find the users with the gold badge in answering, named “Illuminator”, that is given to users that have edited and answered 500 questions (both actions within 12 hours, answer score > 0).

III RESEARCH METHODOLOGY

3.1 *Who are the most active users?(Q1)*

To accomplish this task, attributes such as post type ID, answer count, question title, and creation date from Posts.csv is used. MapReduce framework in MRJob is used to complete this analysis. In the mapper, only the posts with questions are considered by limiting analysis to posts that have post type ID = 1. For all questions, the creation date, title, and answer count is extracted. The year is extracted from creation date and the mapper yields year as the key and the title and answer count as the value. In the combiner and the reducer, the year, title and answer count of the maximum answer count are yielded. The entire program took about half an hour

3.2 Where *do the users that answer the most come from?(Q2)*

Having prepped the data, the final results are determined by using a MapReduce framework on the combined file. Based on the file identification, the mapper yields different keys. If the file is identified as "badges", the mapper yields the user_id as key and badge name as value for those users who badge matches the criteria, in this case, illuminator. If the file is identified as "users", the mapper yields the user id as the key and the location as the value. The location is a user entered value and it is not in consistent format. Some users include cities and states while others do not. Sometimes acronyms such as USA is used while other times they are not. To address this variation, all locations are converted to latitude and longitude of country name using geopy package. From that, the geopy reverse is used to convert the latitude and longitude into standardized location and the country name is extracted out. Therefore, the mapper output is a consistent location that can be easily visualized.

The mapper outputs two different key-value pairs, but the reducer takes advantage of the fact that the user id in badges is the id in the users is the same to output the user id key and location and badge as the final output. This process is summarized in the figure below

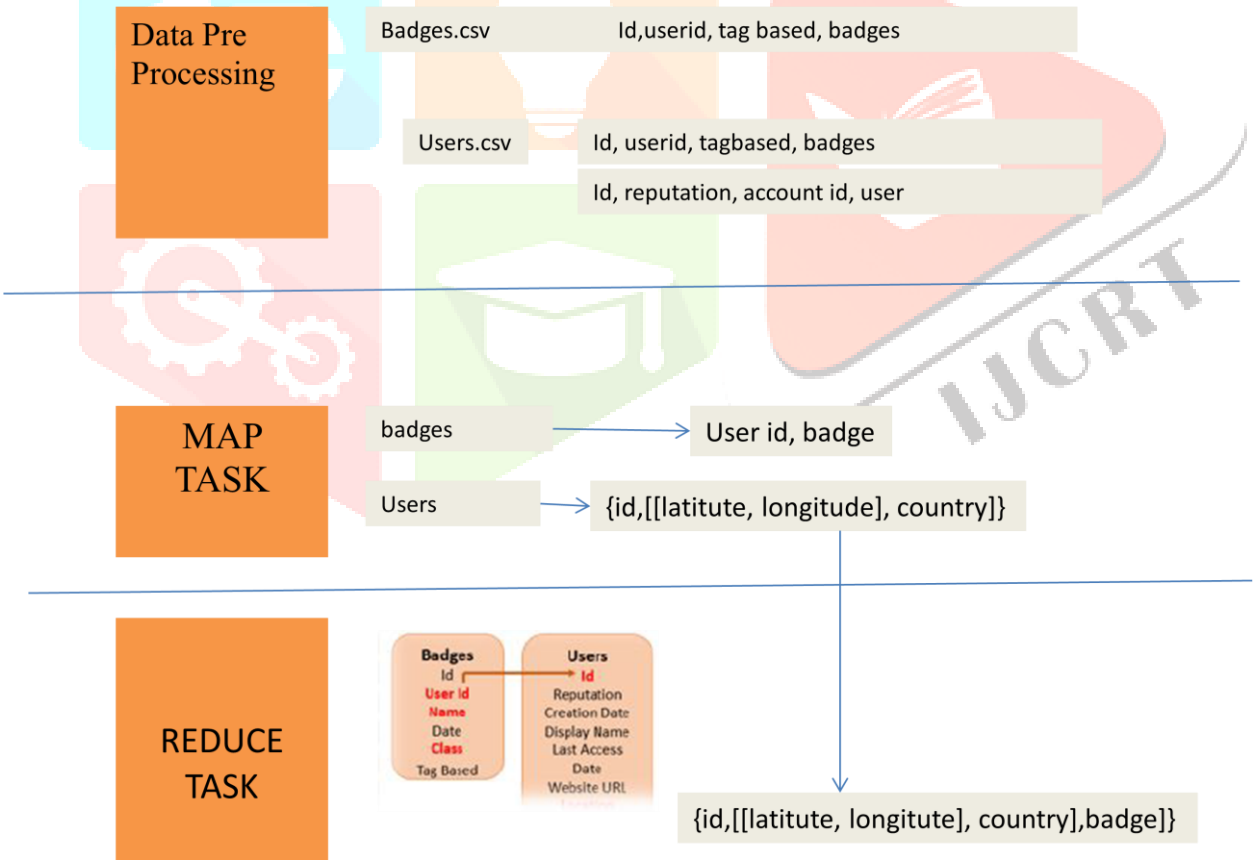

Fig 3.1: Result of the answer providers analysis

III. RESULTS AND DISCUSSION

For the question Q1, It is evident that with time the type of question has also changed. Earlier, open ended questions with very high number of responses seemed normal and acceptable. However, in the recent years, the questions are specific and seem to target error resolution. It is to be noted, the result is missing value for year 2012. This is due to an anomaly carried over from the data cleaning process.

For the Question Q2, from the results of the subset of data, it can be seen that most active answerers come from United States. Due to the low count of users from all other countries represented, it is hardly visible on the map.

.

REFERENCES

[1] Lukas Forer, Enis Afgan, Hansi Weibensteiner, Davor Davidovic, Gunther Specht, Florian Kronenberg, et al., "Cloudflow – A Framework for MapReduce Pipeline Development in Biomedical Research", *MIPRO*, pp. 185-190.

[2] S. Kim Chu, Y. Lin, Y. Yu, G. Bradski, A. Ng and K. Olukotun, "“Map-reduce for machine learning on multicore”", *Proc. Advances in Neural Information Processing Systems 19*, pp. 281-288, 2007.

[3] Alipanah, N., Parveen, P., Khan, L., Thuraisingham, B.: "Ontology-driven Query Expansion Methods to Facilitate Federated Queries," 2010 IEEE International Conference on Service Oriented Computing and Applications (SOCA10), Perth, Australia (2010).

[4] J. Krueger, M. Grund, C. Tinnefeld, H. Plattner, A. Zeier and F. Faerber, "Optimizing write performance for read optimized databases", *ser. DASFAA'10*, pp. 291-305, 2010.

[5] J. Dean and S. Ghemawat, "MapReduce: A Flexible Data Processing Tool", *Commun. ACM*, vol. 53, no. 1, pp. 72-77, Jan. 2010.

[6] Hutto, C. J. & Gilbert, E. E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[7] https://archive.org/details/stackexchange