Programming Language Popularity in Relation to Its Presence on Social Media

Justin M. Peterson, Mohana Priya Ramachandran, Ateeq Rahman

ABSTRACT

Data mining and analysis has opened many doors for researchers and product developers to improve their output and drive results based on a multitude of data sources. "Big Data" has become a coined phrase that describes many concepts, but at the root of each of them is the idea that "Big Data" must be able to produce a relevant result through analytics. Many people only analyze the phrase and think in terms of scale, while the size of the data set may not be the most important factor to consider.

As a case study, take Stack Exchange [4] for example. This service is a Q&A forum commonly associated with software development and computing topics that provides a publicly available data set containing questions, their associated answers, and a number of up votes signifying how relevant a specific answer is. This service can cover any development topic under the sun, but analyzing every question and response will not lead to any conclusive or meaningful results. Instead, pieces of this data set combined with many analysis techniques and other publicly available archived data can validate hypothesis or predict trends in other data sets.

This piece of research aims to determine the most popular set of programming languages through analysis of publicly available social discussions involving these languages. There are several criterion which can define the popularity of a language through techniques such as sentiment analysis of comments, or calculating the frequency of discussion on web forums regarding specific languages.

1. OVERVIEW

There are often emotionally fueled debates between members of enterprise organizations on which language to pick for a new project. While the sentiments of other developers may not be an all inclusive way to pick which language would be best for a given application, going into the decision process with an overview of popularity of a language is a nice starting point to the discussion. The success or failure of a project may not completely depend on the selection of a programming language, but the more information a developer can know up front about their project and the language they decide to use can make all the difference.

There are many challenges that must be tackled to allow the analysis of these sources to be deemed as relevant. The data application developed would need to utilize tools like sentiment analysis and determine a way to include outliers in the popularity rating of a language. For example, regardless of all of the positive appeal given to a language there will always be some developers that provide strong negative reviews because of a specific component of the language. There must be a way to define the specific subtopics of a discussion that are identified as being either positive or negative. There also may be an associated credibility with each source of data. For instance, StackExchange can provide a rating of popularity for a specific user profile that posts a question or answer, but how can the credibility of tweets or Facebook posts be determined other than by sheer volume? Our team will spend a fair amount of time developing and testing strategies to analyze these issues.

A few example data sets that our team has decided to use for this analysis include a crawl of trending twitter topics, tweets, and users from the WWW conference [3], the collection of publicly available StackExchange data, projects on SourceForge, and data scrapes from Facebook [1]. Our team could also look into more broadly ranging data sets such as google search trends, but this research has not currently been done. These chunks of information come from independent sources and will be able to flex our team's knowledge in distributed processing and analysis of large data sets. An example of this process would be to create a dictionary of programming language names and associated strings and parse the massive collection of tweets available to find text strings and associated metadata. These relevant tweets would be stored in our DBMS layer along with information collected from other sources. Sentiment analysis would then be run against these strings and relevant words associated with the defined dictionary to determine an overall positive or negative sentiment for the statement. Statistical analysis that we learn in the course could then be run across this set of result data.

In accordance to the project guidelines, our group intends to divide the project up into a data mining and storage section as well as a data processing and analytics section. The mining and storage section will be done by parsing the existing data sets in their current format and storing them in an aggregate SQL or Key/Value data store. We have not yet determined which store we will use but we will make that

clear in the near future for validation. The analytics section will be done by constructing models in rattle and R to first determine an appropriate measurement of probability, and then an ordering of language popularity based on application context.

2. PROBLEM DEFINITION

Our team plans to utilize three distinct data sources to perform a well rounded analysis. We have decided to use publicly available StackOverflow comments, public Twitter stream data, and github repository metrics to get comments and discussions involving programming languages. There are specific challenges that need to be addressed for each source when data mining. Once all of the information has been collected, each source will be stored in a collection object defined by MongoDB so that distributed queries can be run across each source with limited computing resources. There is still additional work to be done to examine and sift through the data coming from these resources, but the initial mining step is well underway. All code and documentation that has been written is included with this submission.

Once the data is all stored in a central location, the goal is to perform additional market research in order to filter each statement based on criterion that can help our team discover the relationship between a language's social media presence and its popularity in industry. In the project's current phase, our team would like to see how the overall sentiment conveyed by media users produces a rank for each programming language as compared to already existing rating sources, such as the TRIOBE software list of most used programming languages [2]. The team would like to discover if the opinion of developers about a language actually relates to whether or not the language is used in industry.

We are planning to examine customer interaction patterns in StackOverflow, such as how often they sign in and their frequency in asking questions or answering them. In addition we plan to analyze user's demographics and psychographics based on the data collected to build a model out of the data structure. This will help the team to understand user behaviour and tailor our experiment.

The Twitter mining portion of the experiment involves parsing all available tweets using an incoming firehose of data that s returned from Twitter's API. There is a publicly available endpoint which utilizes OAuth requests to generate a live stream of data, filtered by a set of comma separated search delimiters and geographical data. A small whitelist of search terms has been built to test the stream of incoming data and the effectiveness of API's ability to filter based on search strings. These tweets are then run through a series of sanitization which strips out retweets and other pieces of content that may be erroneous in future analysis. These tweets are then stored in a 'tweets' collection in a local MongoDB instance for further processing and sentiment analysis.

The GitHub data portion of the experiment has not yet been parsed, but the data source has been identified and downloaded to the team's local development server. The analysis on public github repository data will follow the same format as the StackExchange data analysis.

3. DATA MODEL DEFINITION

MORE DETAILS TO COME IN PHASE 2 We are planning to use Clustering, Classification and sequential pattern techniques to build our model. We are also planning to use MapReduce Processing since our database of StackOverflow, GitHub, and Twitter content under study is based on non relational databases. In addition, the team will build a statistical model based on overall sentiment in regards to each popular language to identify and extract the information from the collected data.

Anomaly detection might help the team to target outliers in opinion and to identify the overall relevance of a discussion on each media platform. It helps to understand how the results of sequential pattern analysis may look different from anomaly detection perspective.

Association learning helps us to identify the similarities between a user's history and their next search query. If a person searches something related to big data, they may be more likely to produce more queries related to big data. The concept of association learning is what makes the suggestions in stack exchange that appear to the the right of a user's browser. We can reveal the structure of the data based on the patterns detected and can apply them to predict other aspects of data. Hence we can derive our model based on these associations derived from data mining techniques. Algorithm property: We are planning to use either Segmentation or Sequence analysis algorithms Segmentation algorithms help us to divide the data types into clusters based on their common properties whereas Sequence analysis summarizes frequently repeating events or sequences in data and helps us understand the flow of the data like the user flow or the web flow.The later algorithm in case of our indented dataset Stack Overflow helps us understand the user's interest based on his/her patterns and predict what they might search next or which query might assist them to get the answer they are looking for.

4. APPROACHING DELIVERABLES

The team first must work to define criterion used to relate each separate dataset, such as text content of thread posts, tweets, or comments. Once each data source has correctly defined sanitaztion texhiques, the team can move on to model analysis of the data. From here, we will continually reformat our sanitization and collection techniques and increase the number of studies we are comparing to our results to paint a more accurate picture of our data. Our team's current set of deliverables is outlined as follows:

1. Finish collecting all data sources into MongoDB Instance - Week 5
2. Outline and run all initial sanitization techniques on data - Week 6
3. Perform sentiment analysis on datasets and compare to results - Weeks 7,8
4. Final reporting, dataset adjustments, and documentation of results - Rest of term

5. ARCHITECTURE

TBD

6. IMPLEMENTATION

TBD

7. LESSONS LEARNED

TBD

8. CURRENT STATUS & FUTURE WORK

TBD

8.1 Tables, Figures, and Citations/References

TBD

9. REFERENCES

[1] 2005 facebook crawl. `https://archive.org/details/oxford-2005-facebook-matrix`.

[2] Tiobe programming language popularity rating. `http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html`.

[3] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 591–600, New York, NY, USA, 2010. ACM.

[4] D. Posnett, E. Warburg, P. Devanbu, and V. Filkov. Mining stack exchange: Expertise is evident from initial contributions. In *Social Informatics (SocialInformatics), 2012 International Conference on*, pages 199–204, Dec 2012.