Technical Report - Stack Overflow Analysis

Introduction

Stack Overflow is a question and answer website used mainly by professional and ameteur programmers, where users can pose questions that other users may answer. Deserving questions and answers can be given points (positive votes) by the users of *Stack Overflow*, which indicates that the question or answer was regarded as a good one by the rewarder of the point. Our aim was to research the *Stack Overflow* database and be able to generalize some similarities between successful questions on the platform. Unfortunately our group consisted only of two people, Kalle Belmostefa and Lauri Karanko, due to our third member having insufficient time to participate in the course. As a result of continuous scheduling issues with said person, we were able to work on the project in earnest for less time than we had initially wished to as a group.

Datasets used

The analysis begun on the *Google bigQuery* hosted public *StackOverflow*-datasets, on which users are able to run standard SQL-commands upon, or optionally query the database where the datasets are hosted from a virtual machine hosted on *Google Compute*. After some preliminary unsuccessful attempts to create a google hosted *jupyter notebook* solution, we began using the xml-formatted archived versions of *Stack Overflow* from https://archive.org/details/stackexchange. Most of our testing and focus was on the **code review** *Stack Exchange*-dataset, in which the *"Posts"*-table contained the data most pertinent to our research.

Our Aims

We aimed to find some commonalities between successful questions on *Stack Overflow*. We hypothesized that aside from the more obvious factors contributing to a successful question, there would exist some less obvious links between questions that were successful. Factors such as the time when the question was posed and the length of the question would be easy to measure. Our more ambitious aim was to find out whether there were some words that were more likely to garner a positive reaction as *Stack Overflow* questions.

Methods

We used local jupyter notebook setups to analyze the datasets. We decided that the easiest benchmark for whether a question is good was the amount of points it has received, although other factors such as the reputation (people that have some reputation on Stack Overflow can use more features on *Stack Overflow* when writing questions) of the presenter of the question could skew these benchmarks.

To present and wrangle the data, we utilized the data plotting capabilities of *matplotlib, pandas, numpy and scikit-learn*. Initially the data we were interested in the *Posts*-table was in large arrays that needed to be parsed. By writing several helper functions to remove URLs and html, as well as stack exchange tags and punctuation, our data was almost ready to be consumed by whatever natural language processing we were keen to use on it. Nevertheless, despite these efforts, we still had hundreds of thousands of different unique “words” to parse. Removing all numbers, and using a stopword-library, such as the one provided by the *nltk* python module, we were able to remove the rest of the superfluous words in our posts. The last thing to do was to lemmatize or stem the remaining words, aggregating different conjugations and forms of duplicate words under the same keyword.

What we were able to do with the parsed word lists was to use the tf-idf (term frequency–inverse document frequency) method provided by *nltk* to find the most relevant words in the questions. This same word frequency analysis was ran on a subset of the questions that had received at least 5 points to determine whether there was a difference in the words deemed the most important in both cases.

A less computing intensive analysis was performed on whether the time of the day or the length of the proposed questions had a significant effect on the amount of points received.

Due to the similar nature of all Stack Overflow dataset tables, our data wrangling solutions should work on any Stack Exchange dataset despite us mainly working with the previously mentioned *code_review Stack Exchange* archives.

Results

Our results can be found at:

https://d471061c.github.io/StackData

All graphs that were made using the results are also displayed here.

Conclusions

Around 3pm UTC was the best time to ask questions on the code review Stack Exchange. This is also when there is the most questions asked on the *code review Stack Exchange.* There's 4 phases of the day when most questions are posed, these are between 0-2, 6-8, 14-16 and 21-23 UTC. Questions that used around 1000 characters were the most successful. In the English language, this means the questions are between 140 to 150 words. Some words appeared far more often in the lexicon of successful questions, but this can be attributed to those words been attributed to tags that were generally the most popular on the *Stack Exchange*.

We would have needed to refine our methods, and find some suitable machine learning method to interpret the data. We weren't able to use the vast amount of data available to us at its fullest.

We had hoped to do more, but what we have learned is hopefully extendable to the other *Stack Exchanges/Stack Overflow*. Due to the unwieldy amount of data on

Stack Overflow proper, and us abandoning using google compute, we decided to work on the far less sizable code review *Stack Exchange*.

Thank you to the organizers and teachers of this course.

Kalle Belmostefa

Lauri Karanko