**Improving Crowdsourced Documentation:**

**Examining Answers on Stack Overflow**

Scott Crain, Niranjana Gajjelli, and Marc Hudak

Abstract

The goal of our project is to divorce questions from answers and focus on what makes an answer desirable. One assumption that we make is that highly voted answers reflect what people like to see in answers.

1. **Introduction**

As an unofficial source of documentation, Q&A sites such as Stack Overflow are immensely popular among programmers, and that popularity is only increasing as time passes. As of October 19, 2014, Stack Overflow has over 8.2 million questions and 14 million answers. A mere year ago, Stack Overflow had over 5.5 million questions and about 10.5 million answers. Both of these are nearly 50% increases over a single year. Stack Overflow is constantly growing, and shows no signs of slowing down. However, as with all forms of community-driven content, the quality of the material on Stack Overflow is subject to questioning. With the increasing popularity of Stack Overflow as not only a Q&A forum, but as a form of unofficial documentation to those who come after, low-quality content is a very bad thing to have.

The primary forms of quality control on Stack Overflow are the "voting" and "accepted answer" systems. For any given post, be it question or answer, any Stack Overflow user can vote the post either up or down, and these votes are aggregated into a "score" on the answer. The simplest and most obvious use of this is that posts that score higher are more visible. Users are encouraged to participate and vote through the Reputation system. Therefore, generally speaking, a highly-voted post will be a high-quality, useful post as judged by the Stack Overflow community while a low (or negative) scoring post has been judged to be of little or no value. On the other hand, the accepted answer system is only available to question askers. The asker of a question can choose to mark any answer to his question as the "accepted answer", causing it to appear at or near the top of the answer list for his question, along with the highly-voted posts. Practically, this is very similar to the voting system, though there is the key difference that the only person who determines the accepted answer is the question asker, instead of the whole community.

1. **Problem Definition and Research Questions**

Our problem focuses specifically on the answers that have been deemed to be high quality. Specifically, we will be examining answers with a score of at least 15 and the set of accepted answers. These answers will be evaluated against a set of metrics to answer the question of "What makes for a good answer on Stack Overflow?”. The accepted answers and the highly-voted answers will, nominally, comprise two different data sets, but both are being measured in the same way. To this end, our primary goal is to answer the following two questions: “What factors contribute to Accepted Answers?” and “How do the factors that contribute to Accepted Answers differ from those that contribute to Highly Voted Answers?”

1. **Data Collection**

Our first step with this project was to convert the Stack Overflow data dump to a queryable Microsoft SQL Server database. To accomplish this, we used the “Stack Overflow Data Dump Importer” (SODDI), freely available from github user peschkaj. Using SQL Server Management Studio, we found that there were 13,684,117 posts marked as Answers, and these posts were the foundation of our data sets. We found the average (mean) score of all answers on Stack Overflow to be 2 and the standard deviation of scores to be 12.77. Even though the scores aren’t distributed normally (There were 239,445 answers with a score at least 1 standard deviation above the mean, which is roughly 1.7% of all answers), we felt that this cutoff provided enough answers to be meaningful. We also determined that there were 4,596,596 Accepted Answers in the dump by cross-referencing Post IDs and Accepted Answer IDs.

From there, we developed a small program that pulls each answer under consideration from the database and computes the metric set we’re using to examine each post for that answer and export the data to a comma-separated value file for easy examination. The metrics under consideration are “Noun count”, “Verb Count”, “Adjective Count”, “Total Word Count”, “Link Presence”, “Code Snippet Presence” and “Response Time”. Part of speech counting was accomplished by using the Stanford Part of Speech Tagger. Links were checked by examining the answer for an “<a href” bit in the answer, which would denote a hyperlink. Code snippets were similarly checked for by checking for “<code>”, because that HTML tag is used in Stack Overflow posts to denote a code block. Response time was computed by comparing the post time of the question to the post time of the answer.

1. **Research Methodology and Results**
2. **Conclusions**
3. **Related Works**