Predicting coding question quality using Stack Overflow ratings

Slide 2: Overview

- Stack Overflow (SO)
- What is a question?
- Support Vector Machine (SVM)
- Methodology
- Experiments and Results
- Summary
- Demo

Slide 3: Stack Overflow

Stack Overflow (SO) was created by Jeff Atwood and Joel Spolsky, and was released in September 2008. They created it to offer programmers a place where they could ask questions and get answers to the questions. To be able to measure quality, they used votes, where users could up-vote questions they found useful, and down-vote questions that were wrong. Furthermore, a question is not restricted to have only one answer, allowing multiple solutions to be present.

In fact, Stack Exchange (SE), which were released one year later, is built upon the same model. This means that if you develop a system for SO, you could also expand it to cover all of SE.

SO uses gamification to reward the users for their participation. Gamification means that you use game elements in places which normally would not be considered a game. In SO, users can be rewarded in various ways, but the three main elements are votes, reputation and badges.

Votes are used as a measurement of the question (and answers) quality and usefulness, but are only shown on the given post. SO also sorts the questions and answers based on score, with the exception being answers that are marked as accepted. An accepted answer are answers which the questioner found to be correct for their problem.

Reputation and Badges follows the users, where one can compare Badges to achievements in games. Reputation is not only used to show how much you have participated, but it also restrains the amount of freedom you have on the site (e.g. commenting, voting, answering, etc). Reputation can in fact be used as a measurement of expertise, because there is a limit to how much reputation you can earn daily. To draw an example from games, it is not like in World of Warcraft or Tera, where you can grind repetitive quests somewhat endlessly.

Users can only earn up to 200 points of reputation each day, where an answer gives +10, and question +5. That amounts to posting 20 answers, or 40 questions. After reaching the daily cap, the only way users can earn more reputation, is by having their answer marked as accepted or earning bounties. Bounties are a currency system. If a user has a question no one answers, or the answers given does not solve their problem, users can trade parts of their reputation for a solution.

I could not find a good identifier for what a good question is. There is what I would call a bias factor. By this I mean that if a certain amount of people share the same problem, it becomes a good

question. Not because of the question asked, but because of the problem it needs to solve. However, these are in most cases weighed up by using the Wiki feature, namely Community Wiki. Community Wiki are usually added to posts that are considered to be helpful to the community (and you can even search for Wiki posts in the search field).

The bad questions where easier to spot. A common denominator was the lack of effort when asking their questions. For the most code related questions, they added a code snippet, and said "This doesn't work. Why?". Other examples were large code examples showing a lot of code that was not related, or no code at all and just showing the error message.

If you went to your teacher with 50 lines of code, and said "This doesn't work. Why?". Do you think s/he would know what was wrong? In most cases, the obvious reply would be "what have you tried?", "what is the expected output" or "what is the error that you are getting". A lot of code examples were also badly formatted and had syntax error. There were also obvious signs of homework/school related topics, and in one question the code even contained the namespace "assignment".

SO also wants questions to be unique, so duplicates are often down-voted. There are however some cases where this does not happen, and it could therefore be interesting to see in the long term which would be the most viewed.

TODO: Look into *"... how much it would have cost to pay people to rate the questions if you were doing it on a time spend reading and rating*"

Slide 4: Picture of good SO question

"How do you undo the last commit?", +10,493 votes, Community Wiki
In right bottom corner, number is reference to source (last slide)

Slide 5: Picture of bad SO question

"Forcing function to return if false" [locked, closed, off-topic], -154

Slide 6: Picture of Badges on SO

Picture of question badges

Slide 7: What is a question?

Questions can be generalized into either factoid or broad. Factoid questions usually only have a set amount of correct answers, whereas broad questions can have many answers which all are correct.

When using in education, questions are usually used as a learning tool to either help students learn something new, or through examination to evaluate your knowledge. For research, it could be the goal you are after, because you need to define a problem. You cannot just post a bunch of numbers, and say "These are my results". You need to ask the question "What are these results?", "What does these results tell me?", "What problems arise from these results?".

An interesting reverse situation is a gameshow from the early 90's called Jeopardy. The slogan for the show was "We have the answers, you have the questions". In this show, players were presented with an answer, and had to ask the question to which gave the answer. This could also be an interesting addition to learning, because what if the exam was not based on the teacher asking you a lot of questions? What if the exam was based on you asking the teacher questions to show that you understood the curriculum. Would you be able to ask hard enough questions to show that you grasped the curriculum?

The goal of Question classification (QC) is to categorize questions, since in most cases the goal is to find the answer to the question. By categorizing it, e.g. PERSON, LOCATION, DATE, you can reduce the amount of answers that could be related to the question.

You have WH-words, which are usually the first word in the question. Examples are "What, which, when, where, who, how, why". Some of which are harder to classify, because they are not as restrained as the others ("What", "Why", "How" and "Which").

N-grams is simply reducing the full text into fragments. These fragments can either be words or the characters in the word, and the 'N' represents the amount of fragments. One of the more known is the Bag-of-words (BOW), which is also called a unigram. BOW simply takes every word, counts the frequency and ignores the order. Bi-gram takes every second word, which means some order is kept. Meaning that for higher N-grams, the more focused you are on the sentence, the higher the N-gram should be.

Word mapping and processing ...

Slide 8: Support Vector Machine (SVM)

- Good for regression and classification problems
- Main focus is binary classification
- Often used for text classification
- Separates classes by using a hyperplane
- Four kernels:
    - Linear
    - Radial Basis Function (RBF)
    - Polynomial
    - Sigmoid

Slide 9: Methodology

- Data set based on data dump from Stack Exchange Archive
    - Contains XML files based on table content
    - Imported data into MySQL database
    - Imported data from MySQL database into Pandas.DataFrame
- Development: Python 3.5 and Scikit-learn (0.18.dev0)
- Question processing
- Selecting questions and features
- Selecting estimator and parameters for classification

Slide 9: Experiments and Results

- 6 different features
    - Code samples
    - Hexadecimal
    - Homework (synonyms for homework)
    - Links
    - Numerical
    - Tags
    - All features
- 4 different experiments
    - Unprocessed data set vs. all singular feature, and all questions
    - Unprocessed data set vs. all singular features, and question occurence only
    - Unprocessed data set vs. selected set of features only
    - Stochastic Gradient Descent (SGD) as classifier

Slide 11: Conclusion

- Stack overflow as a question quality metric
- Limitations and issues
- Further work
    - Code blocks, Links and Numerical as a feature set
    - Code analysis
    - Sentiment analysis
    - Version numbering

Slide 12: Demo

Show demo (if time)

Slide 13: Thanks for listening

Slide 14: References