Predicting question quality in question answering forums

Antoaneta Baltadzhieva

Anr. 721353

THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ARTS IN COMMUNICATION AND INFORMATION SCIENCES,
MASTER TRACK HUMAN ASPECTS OF INFORMATION TECHNOLOGY,
AT THE FACULTY OF HUMANITIES
OF TILBURG UNIVERSITY

Thesis committee:

Dr. Grzegorz Chrupała

Dr. Sander Wubben

Tilburg University

School of Humanities

Department of Communication and Information Sciences

Tilburg, The Netherlands

July 2014

Acknowledgement

This Master thesis marks the conclusion of my Master's program *Communication and Information science*, master track *Human Aspects of Information Technology* at the Faculty of Humanities, Tilburg University. I investigate which features influence the question quality on the programming question and answering website Stack Overflow and try to predict which terms determine high and low quality questions. The topic kept intriguing and challenging me, and built upon my previous linguistics Master of Dutch Philology.

I really appreciated the efforts of a few people who helped me with this project and would like to express my appreciation. First and foremost I would like to thank my supervisor Grzegorz Chrupała for his help, invaluable comments on early versions of this thesis, and guidance through the writing process. An additional 'thank you' goes to the efforts of my second reader Sander Wubben. In addition, I would like to thank my dearest husband for his overall support and patience through the last year. Final thanks go to my colleagues for their understanding and flexibility.

Abstract

In the current study, I investigate which features influence question quality, as measured by the *number of answers* and *the question score* a question receives, in a programming Community Question Answering website. Also, I attempt to predict which lexical terms determine high and low quality questions. I test the influence of several variables on question quality. Based on existing literature, I include the question tags, length of the question title and body, presence of a code snippet and the user reputation. In addition, I test the influence of terms used to formulate the question. For each of the two dependent variables, I estimate Ridge regression models with an increasing number of independent variables on a dataset of over 1.7 million questions posted on Stack Overflow. More specifically, I divide these questions into a training, validation and test set. I train the model on the training set, determine the optimal regularization parameter using the validation set, and assess the models' predictive validity using the test set. The results indicate that the inclusion of terms in the models improves their predictive power. In a second stage of the research, the significant terms are ranked based on their coefficient value. The terms with the highest and lowest coefficients were semantically analyzed and divided in subgroups to gain a better understanding of the semantic nature of the terms. According to the performed analysis, terms that predict *high* quality are terms expressing excitement, negative experience or frustration, and terms regarding exceptions, or indicate that the questions are posted by new members. The largest groups of terms predicting *low* quality questions is the group containing spelling errors. Also words that mark off topic questions and interjections are an indication of low quality questions.

Contents

1. Introduction

a. Motivation

The web has changed the way people provide, search and share information and knowledge. It became straightforward to submit keywords in a search engine to express a need, and the search engine immediately lists a large amount of more or less relevant webpages from which the user can choose. However, the search results may not provide an exact solution to the user's problem and it may be time-consuming to review all of them, without having a guarantee of finding the desired answer. Community Question Answering websites offer a new opportunity to obtain the desired knowledge in a more rapid and efficient way.

Community Question Answering (CQA) websites provide an interface for users to exchange and share knowledge. The users in such a forum can be divided in three groups: 1) users who only ask questions, 2) users who only answer questions and 3) users who ask and answer questions (Adamic, Zhang, Bakshy & Ackerman, 2008). The user asking a question lacks knowledge of a specific topic and searches for an expert on the same topic to provide the desired knowledge. In this way, an asker is querying a topic and the experts providing the knowledge about this topic are the source of information, thus replacing other sources like documents or databases. Although the idea of receiving a direct response to a certain information need sounds very appealing, CQA websites also involve risk because the quality of the provided information is not guaranteed. An important difference between user-generated content and traditional content is the range of the content quality: user-generated content shows a higher variance in quality (Agichtein, Castillo, Donato, Gionis, & Mishne, 2008) than traditional content (Anderson, 2006). The quality distribution vary from very high to very low.

Stack Overflow is an example of a question answering community with a question and answer (Q&A) interface in the field of computer programming where programmers can ask and answer programming related questions**.** It is a free site where everybody can ask and/or answer questions and the answers are voted according to the asker's satisfaction[1]. The website is building a library by storing all posted question/answer pairs. The asker can add tags to her

[1] http://stackoverflow.com/

question to annotate to which subject it is related. Clicking on a tag will provide a user with a list of other question-answer pairs about this particular subject[2]. Questions, answers and edits can be voted to indicate how helpful they were to the user. The difference between positive and negative votes determines the user's reputation. Answers are sorted based on the vote-based score. The asker can mark the answer that worked best for him as "accepted" which is visualized by a checkmark. Stack Overflow's goal is to create a high-quality library by collecting high-quality questions combined with the best answers to these questions. To achieve this, they allow their users not only to post questions or answers but also to edit them. By adding a comment a user can also clarify her question or answer.

Despite the encouragement of the website and the offered opportunities to maintain and preserve the quality of the content, a lot of questions on Stack Overflow are not answered. With the increase in popularity of Stack Overflow, not only the number of questions and the number of new members increased, but also the number of unanswered questions. According to statistics from 2012, approximately 45 questions per month remained unanswered (Asaduzzaman, Mashiyat, Roy, & Schneider, 2013). By March 20, 2014, the number of unanswered questions was 752,533 out of 6,912,743 (approximately 10.9%). Interestingly, the fact that those questions are not answered is not caused by users not have been seeing them. In fact, unanswered questions are seen 139 times on average (Asaduzzaman et al., 2013). It is not obvious why a certain question receives an answer whereas other questions do not. Also, it is not clear whether the question characteristics that determine whether a question receives an answer also influence two other outcome variables that could be seen as indicators of question quality: how many answers a question obtains, and the score awarded to the question. In this research, I will evaluate the features of questions in Stack Overflow, how they influence the two indicators of question quality – the number of answers a question receives and the question score - and will make an attempt to predict these outcome measures for newly posted questions.

b. Research questions

Although several studies have been conducted on the content quality of question answering websites, little attention has been paid to the content features of questions. The few existing

[2] http://stackoverflow.com/about

studies have mainly focused on the assigned tags, assuming that they represent the question topic. However, to the best of my knowledge, no work has addressed the exact terminology used for formulating the questions. Moreover, no single study has looked at the two measures of question quality to determine to what extent these are influenced by question features. To fill this gap, the main aim of this thesis is to differentiate between high quality and low quality questions and to predict how likely it is for a new question to be answered, how many answers may be expected, and what score the question will obtain. I therefore formulate my research question as follows:

1. Which question features presumably influence the probability of obtaining an answer, the number of answers, and the question score?
2. To what extent do these question features influence the number of answers and the question score?
3. Based on these features, can we make predictions about the number of answers a question will receive and the question score it will obtain from the community users?

To answer the first question, in Chapter 2 I discuss the existing literature. Based on the literature the following features are considered: the reputation of the user, the existence of a code snippet in the question, the length of the question title and body, the tags chosen and the terms used to explain the information need.

For the second research question, I perform two multiple linear regression analysis to determine to what extent the features determine the number of answers and the score the users assign to a question. I thus perform two analyses: first one where the dependent variable is the number of answers a question received, and a second analysis with the question score as dependent variable.

To answer the third research question, I use the parameter estimates from the multiple regressions to predict the outcome measures for questions that were not used in the estimation phase.

c. Contribution

i. Practical contribution

Question quality is important for question and answering platforms because having appropriate question-answer pairs attracts more users and improves platform traffic (Mamykina, Manoim, Mittal, Hripcsak & Hartmann, 2013). The aim of the asker is to obtain the information she is searching for and the aim of the person providing an answer is to give a satisfying answer in order to increase her reputation. In StackOverflow there are a lot of questions with differing numbers of answers with a large number range. Moreover, it is observed that low-quality questions receive low-quality answers (Agichtein et al. 2008). Low-quality questions also affect the user's experience on question and answering website. Furthermore, high-quality questions improve the entire question and answering platform as these questions are more appealing to users to share their knowledge and in this way to improve the overall platform knowledge. Finally, high-quality questions improve question retrieval and question recommendation in question and answering websites (Agichtein et al. 2008). In this research I focus on some question features to investigate their influence on two measures of question quality: the number of answers, and the question score.

In a good forum there should be a supply and demand balance – the aim of an asker is to receive a good answer to her question and the aim of a person providing an answer is to give a satisfying answer in order to increase her reputation. Therefore, it is essential for Q&A forums to better understand what characterizes questions that are more appealing to be answered. A well-formulated question will increase the answerer's willingness to answer and will help her to give an appropriate answer which, at the same time, will increase the asker's satisfaction. It can also increase the question score and respectively the user's reputation which would motivate her to increase her participation and privileges on Stack Overflow. Seen from the asker's side, not receiving an answer, or only a limited number of answers, is not only disappointing, but may also be of educational or professional disadvantage. What is more, even if a question finally receives an answer, the longer it takes to get an answer, the more likely that the answer quality will not be satisfactory (Anderson, Huttenlocher, Kleinberg, & Leskovec, 2012).

ii. Academic contribution

Several studies have focused on the quality of *answers* on Q&A websites (e.g. Lou, Lim, Fang, & Peng, 2011; Jeon, Croft, Lee, & Park, 2006; Suryanto, Lim, Sun and Chiang, 2009). Despite an increasing interest in the *question* quality in recent years, existing research has paid only limited attention to this topic (for a detailed overview see chapter 2). With respect to Stack Overflow, existing studies have focused on a number of question features. Following previous research, I expect that a code snippet in a question makes the probability of receiving an answer higher, not only in review questions, but in general. The length of the question and of the question body may also play an important role. Interestingly, results from existing research are mixed: whereas Yang et al. (2011) find that short and long questions are more likely to be answered, Asaduzzaman et al. (2013) find that too short questions have a low probability of obtaining an answer. To the best of my knowledge, this research is the first to analyze the terms used in the posted questions and to explore to what extent they can predict the probability of receiving an answer. The results indicate that the models have the best predictive power when the terms are included. The analysis of the terms show that terms expressing excitement, negative experience or frustration, and terms regarding exceptions or posted by new members account for high quality questions. The largest groups of terms predicting low quality questions is the group containing spelling errors, off topic words and interjections.

iii. Outline

The outline of this thesis is as follows. Chapter 1 gives an introduction on the question and answering platforms and emphasizes the importance of question quality. The question features which will be analyzed are briefly introduced. Chapter 1 also defines the research questions. Chapter 2 provides the theoretical framework, containing an overview of previous research on question quality in question and answering websites. It will also explain why the question features were chosen for the research. In Chapter 3, the method of data collection and techniques to analyze and predict the number of answers and question score are presented. In Chapter 4, the outcomes from the linear regressions are presented along with the out-of-sample prediction statistics. Chapter 5 presents the discussion and indicates how the findings of the current research compare to previous work. Chapter 6 provides a conclusion of this research. This last chapter also discusses the limitations and possible improvements.

2. Background

a. Question quality

As there is a large number of CQA websites, it is important for a CQA website to provide high-quality content to distinguish itself from other websites. The importance of high-quality content in community-driven question and answering websites has been recognized and investigated in several studies. Importantly, Agichtein et al. (2008) have shown that there is a correlation between the question quality and answer quality, i.e. question quality will influence CQA service quality. Agichtein et al. (2008) found that good answers are more likely to be given in response to good questions. Similarly, bad answers appeared in response to bad questions. According to the definition of Li, Jin, Lyu, King, & Mak (2012), high quality questions are expected to draw greater user attention, to have more answer attempts and to become the best answer within a short period of time. High-quality questions thus help to improve the CQA website's popularity as they, on the one side, contribute to efficient problem solving, and on the other side, enrich the community knowledge.

Different studies employ different definitions of question quality. As a measure of question quality I consider the response of the community (Anderson et al., 2012.). In this research I focus on two measures of community response question quality: the number of answers and the question score.

As a first measure of question quality, I consider the number of answers a question receives. In a domain specific QA website users share expertise knowledge, mostly in the form of a question-answer pairs. These question-answer pairs are saved on the website and usually ranked by search engine which makes them retrievable and valuable for future information needs (Anderson et al., 2012). A question has a long term value when it draw the users' attention long into the future after it was posted. Research has shown that the number of answers is the most significant feature to predict the long term value of a question together with its answers set (Anderson et al., 2012). A larger number of answers results in more views and provides higher value in the long run (Anderson et al., 2012). The number of answer is a direct feedback on the usefulness/quality of the question. If the users assess it as being off topic or for some other reason inappropriate for the question answering community, the users will be less likely to provide an answer.

Other research focuses not on the number of answers but on whether a question received at least one answer (Yang, Bao, Lin, Wu, Han, Su & Yu, 2011). Similarly, Asaduzzaman et al. (2013) conducted a qualitative study on unanswered questions. Although one might think that questions remain unanswered because they were for some reason not discovered, Asaduzzaman et al. found that unanswered questions were on average seen 139 times. In this study, to predict whether a question will receive at least one answer, I will investigate the number of answers a question received – it would be 0 if the question were of no interest to the users and would increase gradually depending on the significance of the question to the community.

Third, to indicate whether a question has been valuable for the question and answering community, it can be rewarded by voting it up. Equivalently, if the question was of low quality or useless to the community, the asker can be punished by voting her question down or even deleting it from the website. In general, answered questions on Stack Overflow have higher scores compared to unanswered questions (Saha, R.K., Saha, A.K. & Perry, 2013). Hence, the question score is an appropriate feature to measure how the community users assess the question quality. A question can be voted up or down by using the up or respectively down arrow on the left side of the question. The question score consists of the sum of the upvote count and the (negative) downvote count. As I am interested in the response of the question answering community as a whole, I will focus on the end score of the question, not on the separate up- and downvotes.

As both the question score and the number of answers are considered quality determiners, one would expect that a question with a high score receives a lot of answers. That would be the case when a question is found very interesting and valuable to the community and there were enough experts to answer it. Also if a question was, for example, not appropriate for the programming domain, it may not receive an answer and get a lot downvotes. However, the question score and the number of answers may not necessarily correlate. A question may address a new development that is very interesting to the programmers community but at the same time also very difficult to answer as there may be not enough experts familiar with it. Such a question may receive no answers but a lot of upvotes. If however a question was too easy or posted previously it may receive answers, but not evaluated high as it does not contribute to the question answering website.

A number of other measures of question quality have been used in the literature. LaToza, & Myers (2010) conducted a survey among professional software developers to identify code-related questions they find difficult to answer. Although I will not explicitly measure the difficulty levels of questions, it can be expected that difficult to answer questions will obtain no or few answers. Correa, & Sureka (2014) are also interested in the content quality of Stack Overflow, but instead of investigating the features characterizing well-formulated questions, they concentrate on the features that describe low-quality questions. In Stack Overflow, if questions are off topic or of poor quality, they can be deleted by Stack Overflow moderators, experienced users with high reputation, or by the user who posted the question. As a deletion of a question is direct feedback regarding its quality, Correa et al. (2014) aim to find out what defines a question that is considered bad enough to be deleted as well as how long it takes to remove it. The statistics revealed that for most of the deleted questions it took a long period of time to receive the first delete vote - approximately eighty percent of questions after one month, and half of the questions after six months. They also find that eighty percent of deleted questions received one delete vote and 14 percent received 3 delete votes. If the question was deleted by the asker herself, it was removed much faster than by a moderator. The feature that has the biggest influence on the deletion decision is the question score. Correa et al. (2014) show that eighty percent of the deleted questions have a zero score. Hence, although I do not explicitly focus on question deletion as a measure of question quality, the second measure that I consider – the question score – will be highly related to the probability of a question being deleted.

b. Features determining question quality

To classify the features which have influence on these outcome measures, I divide the question attributes in two groups: question-related and asker-related attributes. The group of the question related features is represented by the features *tags, terms, question title and question body length* and *the presence of a code snippet*. Regarding asker-related features, I will take the reputation of the user into consideration. Since one of the goals of this paper is to predict the number of answers and the question score, I focus on features that relate to information that is available at the moment a question is posted, i.e. I leave out those features that contain information that only becomes available once we already know whether and how many answers a question received. The reasoning behind this choice is that features which are not available at the moment of the posting cannot help the asker to improve his or her question (Chang, Schiff, & Wu 2013; Correa et al., 2014).

i. Question-related features

Tags and terms

In Stack Overflow, the asker can add tags to her question to indicate to which topic(s) the question is related. Intuitively, one would expect that some question topics will elicit more answers than others, just because more people might be working on a certain topic, i.e. there will be more potential answerers available. Although tags may potentially differentiate between the number of answers, Saha et al. (2103) conclude that the large number of unanswered questions cannot be explained by a lack of sufficient experts for certain topics. They considered the assigned tags as representative topics and investigated tags used for unanswered questions but not for answered questions. They found 274 unanswered topics linked to only 378 questions in total. The number of questions with these specific tags is very small compared to the total number of unanswered questions which would indicate that there is at least one expert for each tag/topic. However, as users mostly assign several tags to a question, covering very general to specific tags, the large number of unanswered questions cannot be explained by a lack of experts (Saha et al., 2013). Correa et al. (2014) also analyzed tags to investigate the topics of questions in Stack Overflow. They found that approximately ten percent of the tags found in deleted questions were not present in closed or regular questions. These questions, tagged for example as homework, job-hunting and polls, are beyond the interests of the programmer community.

Saha et al. (2013) and Correa et al. (2014) assume that tags are representative of the actual question topics. Asaduzzaman et al. (2013), however, state that incorrect tagging is one of the characteristics of unanswered questions. Nasehi, Sillito, Maurer & Burns (2012) investigated, among others, the relationship between characteristics and the question type in Stack Overflow data. They describe question types based on two dimensions – the question topic and the main concern of the asker. The former is described by the technology or construct the user is asking about, and the latter dimension concerns the problem the asker wants to solve. Nasehi et al. (2012) consider the following question types based on the problem of the asker: debug/corrective, need-to-know, how-to-do-it, seeking different-solution. Nasehi et al. found that the answer attributes are likely to be determined only by the second question dimension, the main concern of the asker. Correa et al. (2014) observed that a high percentage of author-deleted questions are marked as too localized and off topic, and that a high percentage of moderator-deleted questions are marked as subjective and not a real question. These results indicate that question topics, i.e. tags, may either be incorrect and/or may not be fully informative of the likelihood of receiving an answer, the number of answers, or question score.

Therefore, a number of recent studies tried to infer question topics from the natural language used to formulate the questions. Wang, Lo & Jiang (2013) assume the number of the topics to be equal to five and use Latent Dirichlet Allocation (LDA) (Blei & McAuliffe, 2003) to find latent topics. They manually label each of these topics to: user interface, stack trace, large code snippet, web document, miscellaneous. Table 1 shows the topics with the representative key words. Wang et al.(2013) found that the category of miscellaneous topics which consists of many different kinds of questions, hold the largest number of questions. The second largest category is the web document topic, followed by large code snippet, stack trace and user interface.

Table 1. Topics and related words

Topic	**Words**
User Interface	view, image, button, etc.
Stack Trace	java, error, org, server, etc.
Large Code Snippet	code, string, new, object, class, etc.
Web Document	href, page, html, php, etc.

Similarly, Yang et al. (2011) infer question topics from natural language using supervised latent Dirichlet allocation (SLDA) for classification (Blei and McAuliffe, 2007). They focus on questions in Yahoo! Answers and set the number of topics to 50. They discovered that, unsurprisingly, the topic with the lowest probability of remaining unanswered is a seasonal topic (questions were crawled on April 1st and the most answered questions were about “April Fools”). The second and third most answered questions are related to pets and food, respectively.

In this thesis, I use both observed topics, i.e. tags, as well as information from the questions’ natural language formulation. Instead of extracting latent topics as proposed by Yang et al. (2011), I will focus on the presence of words. Because there are 36,643 different tags on Stack Overflow, I assume that they will cover a pretty large number of possible topics. To avoid similarity between observed tags and latent topics, I instead concentrate on the presence of certain terms in the natural language of the questions.

Length of the question

Yang et al. (2011) found that the top 10% short and the top 10% long questions have the highest probability of obtaining an answer, while the medium long questions were less likely to be answered. They explain this phenomenon by noting that reading and answering a short question can be accomplished in a very short time. Long questions are mostly expertise-related and need more explanation. These questions attract more users with the same interest and are therefore more appealing to be answered. Yang et al. (2011) assume that medium length questions are less interesting and unnecessarily long which makes them less likely to receive an answer. In contrast, Asaduzzaman et al. (2013) found that too short questions are very likely to remain unanswered. Those questions may miss important information, be too vague or unclear. Also too time-consuming questions are not very attractive for answerers. Saha et al. (2013) also explored the effect of question length on question quality. They found that for answered questions, the minimum length is 5 characters and the maximum length is 48,258 characters[3] (*M* = 1,079, *SD* = 1,389); for unanswered questions, these numbers are 19 and 35,588, respectively (*M* = 1,300; *SD* = 1,845). Their results show that although unanswered questions have larger length, both classes have the same probability of receiving an answer. In the ranking list of importance for differentiating unanswered from answered questions the attribute “question length” gets the same place for both classes. Correa et al. (2014), finally, found that compared to closed questions, deleted questions had a slightly higher number of characters in the question body. Existing literature thus does not provide a consistent answer to the question of whether and

[3] Although from the article it is not clear what unit is used to measure the length, I assume Saha et al. (2013) calculated the number of characters.

to what extent question length influences question quality. Further, it is not clear whether users mainly look at the length of the question title or the question body in deciding whether to answer the question or not. In previous work, question length and question body length are never analyzed separately. Therefore, I explore the effects of both to see if the results point in the same direction.

Presence of a code snippet

Stack Overflow is dedicated to programming-related questions. Treude, Barzilay & Storey, (2011) manually analyzed a few hundred questions and assigned them to ten different categories. Their goal was to investigate which questions are answered well and which remain unanswered. Using qualitative and quantitative data from Stack Overflow, they distinguished ten question categories – how-to, environment, discrepancy, error, decision help, conceptual, review, non-functional, novice and noise. They found that review questions had a high answer ratio as they often contain a code snippet and may have more than one possible good answer. The presence of a code snippet will not only elicit more answers to review questions; also other types of questions may benefit from the extra information provided by a code fragment. Asaduzzaman et al. *(*2013), for example, found that program specific questions are very hard to answer if no code fragment or detailed explanation is included. Correa et al. (2014) analyzed deleted and closed questions. Question that are extremely off topic, have very poor quality or have no activity after a long period of time, are deleted. A user can delete her one question when it has not received any answer or upvotes[4]. Closed questions also indicate low quality and are questions which are considered duplicate, subjective, off topic, too localized or not a real question (Correa & Sureka, 2013). Similarly to the research of Asaduzzaman et al. *(*2013), they found that deleted questions had a lower percentage of code blocks compared to closed questions. Interestingly, the presence of a code snippet may have adverse effects as well. A user may not receive an answer if the code is hard to follow or if other users are able to understand the code but cannot see the problem (Asaduzzaman, et al., 2013.). The possible adverse effects may explain why Saha et al. (2013) found the presence of code to rank only ninth in terms of importance for differentiating between answered and unanswered questions.

[4] http://stackoverflow.com/help/deleted-questions

ii. Asker-related features

With regard to asker-related features, I consider the asker's reputation as a feature that influences question quality metrics. The users' reputation scores are built on their participation on the question answering website. It has been shown that the experts, i.e. users with high reputations, do not only provide an essential contribution to answering websites in general, but they also provide the most helpful answers (Welser, Gleave, Fisher, & Smith, 2007*;* Pal, Harper, & Konstan, 2012). On Stack Overflow, reputation can be gained when user's question or answer is voted, when an answer is accepted as the best answer or by receiving a bounty etc. A bounty can be assigned to a question when the question is not receiving answers and for that reason a user want to draw the attention to this question. After the bounty period ends, the user placing the bounty, can award it to the user providing the best answer[5]. To encourage users to submit and maintain high quality content on the website, Stack Overflow rewards upvotes on answers more than upvotes on questions. In addition, high reputation users have more privileges in site management and receive more bonuses than regular users. The most reputation points are scored when a user's answer is accepted as the best answer, when it is upvoted or when the answer has received a bounty. Anderson et al. (2012) show that users build their reputation mainly by receiving upvotes for their answers and not by asking questions themselves. Nevertheless, when high-reputation users will post a question, I expect these questions to be of higher quality, because high-reputation users have more experience in answering questions themselves, i.e. they would be better skilled in what topics are more popular among the community and how to formulate their questions in order to receive the required answer. New users would be, on the contrary, less experienced in what to ask and how to ask it. Indeed Saha et al. (2013) found the asker's reputation to be one of the most dominant attributes to distinguish between answered and unanswered questions The minimum reputation score for answered questions is 1, the maximum score is 465,166 (M = 1,886; SD = 7,005). For unanswered questions the minimum score is 1, the maximum score is 223,117 (M = 579; SD = 2,586). This result is in line with Yang et al. (2011) who define the asker's reputation by the number of resolved questions posted, the number of answered questions and the number of answers that were indicated as best answer. They showed that users with a larger question and answering history were more likely to receive an answer

[5] http://stackoverflow.com/help/bounty

than new users. Similarly, Asaduzzaman et al. (2013) found that the number of unanswered questions decreases with an increase in the asker's reputation.

3. Methodology

a. Data set

The dataset consists of JSON files extracted from Stack Overflow containing questions in the period between 31 July 2008 and 9 June 2011. Within this time period, 1,713,400 questions were posted. Out of the total number of questions, 126,227 remained unanswered (7.37%). Each question contains information about the question itself, such as title, body, upvotes, downvotes etc., and about the question owner, e.g. registration status, reputation, name, id etc. In this research I am only interested in the variables as described below.

b. Variable definitions

i. Dependent variables

As described in Chapter 2, I use two measures of question quality: the number of answers and the question score. The number of answers and the question score are integer outcome variables. For the former, simply the number of answers that each question received is taken into account. The question score is a direct indication of whether a question obtains a positive assessment from the Stack Overflow community as it represents the sum of the upvotes and downvotes given by the users. In my analysis, I leave without consideration the separate up and down votes and only analyze the end score of the question.

ii. Independent variables

Tags and terms

For the prediction task two content features are included – the assigned tags and the terms used to formulate the questions. Each tag is represented as a binary variable. It receives the value of (1) if it is assigned to a question and a value of (0) if it is not assigned.

Term extraction is the process of automatically generating a list of terms which is part of the prediction task. For this purpose, I use the Scikit-learn module in the Python programming language. The module provides utilities to turn textual symbol sequences into numerical values that are used in the regression models. The terms are analyzed as the number of occurrences in the question title or question body where a term receives a value of 0 if it does not occur and otherwise the value of the number of occurrences, e.g. 3. In order to extract numerical information from text content, first a tokenization process takes place, chopping character streams into pieces while throwing away characters such as punctuation (Manning, Raghavan, & Schütze (2008). The CountVectorizer from the Scikit-learn module is used to exclude stop words. Stop words are function words, which are normally filtered out from the vocabulary prior to natural language processing, because they are of little value in finding documents matching a user's information need (Manning et al., 2008). The CountVectorizer filters the stop words and returns an appropriate stop list[6]. Further, in this study I only include tags that appear in at least 20 questions and terms that appear at least in 50 questions. Results based on tags and terms that occur seldom are likely to be spurious and I assume that such tags and terms would not have strong predictive power anyway. Stemming is sometimes used to reduce inflectional or derivationally related forms of a word, such as *programmer*, *programs* and *programming*, that have similar meanings to its base form, its stem. With other words, stemming is a process of removing the ending of the words or derivational affixes of the words. In the current study, I do not apply stemming as preliminary analyses indicated that it did not improve the model fit.

[6] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

Question title and question body length

Length of the question title and the question body are considered as two separate independent variables. Community users have to read and analyze them both to fully understand what information the asker is seeking. Furthermore, in some views the interface only shows the question title. A user needs to click on the question title to view the question body. The number of terms in both parts are counted and the totals are used as independent variables. As the code snippet is considered a separate independent variable, it is excluded from the question body length. Further, all words, including less informative words such as stop words, are counted as they all contribute to the question length and might result in the unwillingness of the community users to answer the question if it is for example too long.

Presence of a code snippet

In this work I consider only blocks of code, marked by <pre><code>, and no inline codes (marked by <code>). Most of the time, the inline code refers to tagged method names or other programming language terms and not to real code snippets. The presence of a code snippet is represented as a binary variable. It receives the value of (1) if at least one code snippet is available in a question and a value of (0) if it is not.

User reputation

The quality of the user's interactions on Stack Overflow is measured mainly by voting – users gain reputation points when their questions and answers are voted up when a user's answer has been accepted, i.e. when the asker indicates a specific answer as the accepted answer, by an accepted suggestion, for editing an answer, etc.[7] In my research I use the current total reputation score.

[7] http://meta.stackoverflow.com/questions/7237/how-does-reputation-work

iii. Variable overview

Table 2 gives an overview of all independent variables and their definitions.

Table 2. Independent variables

Independent variable	**Definition**
Tags	The tags are represented as a Boolean value.
Terms	A count variable of how often a term occurs in the question or question body.
Question title length	The number of words of the question title. Inline code is not included.
Question body length	The number of words of the question body. Inline code and code blocks are not included.
Code snippet	Whether a code block is present in the question body. Only blocks of code, marked by <pre><code>, are considered and no inline codes (marked by <code>).
User reputation	User reputation score.

c. Descriptive statistics

Tables 3 and 4 provide an overview of the data and descriptive statistics of key variables, respectively. Surprisingly, the amount of questions without a code snippet is substantially larger than the number of question that do contain a code snippet. To analyze the tags assigned to a question and the terms used in a question and question body, I take into account only tags that were assigned to at least 20 questions and terms that occur in at least 50 questions in the dataset. As explained before, results based on tags and terms that occur seldom are likely to be spurious and I assume that such tags and terms would not have strong predictive power anyway.

Table 3. Data overview

Data item	Count
Questions total	1,713,400
Questions unanswered	126,227
Code snippet 1/0	792,822/920,578
Terms after normalization	36,865
Tags after normalization	11,613

Table 4. Descriptive statistics

	Mean	Standard deviation	Median	Min	Max
Number of answers	2.28	2.54	2	0	636
Question score	1.44	5.31	1.00	-49	2330
Question title length	8.27	3.71	8.00	1	39
Question body length	91.74	87.43	72.00	1	18716
User reputation	1600.40	5552.40	301.00	1	308526

From table 4 it is obvious that all variables, except the length of the question title, are skewed to the right. As extreme values can influence the descriptive statistics values, the variables are normalized by the logarithmic transformation or percentile method before use in the linear regression. I discuss the normalization in more detail in the next paragraph. The number of answers is very large for some questions, but most questions receive a small number of answers (M = 2.28 and SD = 2.54). It is interesting to see that despite the fact that the question score can reach a very high value, on average a question receives a score of 1.44. This is possible because questions can be downvoted and receive a negative score. Another interesting observation regards the minimum and maximum value of the variables *question body length* and *user reputation*. In both cases, extremely large values cause the skewness to the right. Interestingly, some question titles and bodies consist of only one word. These may be questions where only a code snippet was posted. The high mean value of the user reputation suggests that many Stack

Overflow users have a high user reputation. As it has been shown that there is a positive relationship between the user reputation and how fast the user replies to a question (Anderson et al., 2002), it can be concluded that Stack Overflow askers are active users in the community.

As mentioned above, I normalize the variables used in the regression analysis to reduce the influence of extreme values. The independent variables *title length*, *body length and user reputation* are normalized by the logarithmic transformation using the natural logarithm. With regard to the dependent variables, I use percentile normalization. As *question score* and *number of answers* can have non-positive values, the logarithmic transformation cannot be performed. The percentile normalization sets all values above the selected percentile to the value of the selected percentile. For the current analysis, the 99.9^{th} percentile is used, thus setting all values above this percentile to the highest value within the 99.9^{th} percentile. This way most values remain unchanged for the analysis. The resulting descriptive statistics of the dependent variables are shown in Table 5. As can be seen, the maximum values of number of answers and question score are now 12 and 19, respectively. The mean values do not change to a large extent, however, the standard deviations drop dramatically. The analysis will be performed on the transformed dependent and independent variables.

Table 5. Descriptive statistics of the dependent variables transformed by percentile normalization

	Mean	Standard deviation	Median	Min	Max
Number of answers	2.242	1.869	2	0	12
Question score	1.331	2.446	1.00	-49	19

Further, I calculate the relationships between the independent variables to see whether there is multicollinearity. Due to space constrains, I exclude the variables *terms* and *tags*. Only the relationship between (1) the title length and the body length and (2) the relationship between the body length and the code snippet exhibit a low correlation, $R_1 = 0.10$ and $R_2 = 0.08$. The first relationship implies that users who tend to give much detail normally do so in both parts of the question, or they have a more complex information need that needs detailed explanation in the question title and the question body. The second relationship could be simply caused by a bigger chance of a presence of a code snippet in a larger texts – the larger the text, the bigger the chance of a code snippet that could help to illustrate part of the text. The other variables are not significantly correlated with each other.

d. Method of analysis

This section discusses the techniques used to test which factors influence question quality. To predict how the question is evaluated by the users and to what extent the question features influence the number of answers and the question score, a multiple linear regression model is used. Multiple linear regression is an extension of simple linear regression in which instead of one, there are multiple predictor variables where every additional predictor variable has its own coefficient. The prediction is then made by a combination of all predictor variables multiplied by their coefficient. The expected relationship is a linear function of the independent variables (Field, 2009):

$$y_i = \beta_o + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi} + \varepsilon_i$$

In this research, for question i, y_i represents the dependent variable *question score* or *number of answers* received, the βs represent the coefficients of the predictor variables x_i and ε_i is the difference between the predicted and the observed value of the outcome variable y_i, which is assumed normally distributed. To predict question quality, I use the question-related and the asker-related features described in sections 3.b.i and 3.b.iii. The aim of the analysis is to assess whether the independent variables *tags, terms, question title* and *question body length, the presence of a code snippet* and *user reputation* explain and predict the outcome variables.

Ordinary least squares regression minimizes the residual sum of squares between the observed responses in the dataset, and the responses predicted by the linear approximation. The technique of ordinary least squares regression is known to be not very accurate when predicting future responses and investigating the relationship between the response variable and the predictor variables (Van der Kooij & Meulman, 2008). In these cases regularized regression models are preferred. Ordinary least squares regression may have highly variable estimates of the regression coefficients when there is multicollinearity or when the number of predictors is very large in connection to the number of observations (Van der Kooij et al., 2008). Multicollinearity implies that predictor variables in a multiple regression are highly correlated. In programming languages, a lot of terms appear together. For example, if a user is searching for information on how to concatenate arrays, then one would expect that the terms *concatenate* and *array* will occur often together, hence leading to multicollinearity. In this study, all terms used in at least 50 questions are considered as independent variable. As their number is extremely large (36,865) and in order to avoid overfitting, a regularized regression model, namely Ridge regression (Hoerl & Kennard, 1970a, 1970b), is used. Ridge regression applies a penalty to the sum of the squared values of the regression coefficients which shrink the coefficients towards zero. In Ridge regression the coefficients shrink towards zero, but never become zero, which means that all predictors remain in the model. Applying a penalty results in lower expected prediction error because it reduces the estimation variance (Hartmann, Van der Kooij & Zeeck, 2009).

To prevent overfitting and to be able to assess the predictive validity of the models described below, the dataset is split in three subsets – a training set, a validation set and a test set. The training set consists of 60% of the data and is used to build up the models. For the validation set 20% of the data, that is different from the training set, is used to optimize the regularization parameter α for each model. To find the optimal ridge parameters, several values are tried in increasing order. The α value that reduces the Mean Squared Error (MSE) of the validation set the most is chosen as the optimal parameter. Finally, the obtained coefficients, given the optimal regularization parameter, are applied on the test set to assess the predictive validity of the models. Because the validation set and the test set are not used in any way when training the model, it is expected to provide a more objective measure of the performance of the models for predicting question score and number of answers than the results obtained from the training set.

To build the multiple regression models, the Scikit-module in the Python programming language is used. Each question is represented as a vector consisting of the asker-related and the question-related features. Both variables, tags and terms, are built as matrices where each column represents a tag or term and each row includes the tag or term frequency.

To investigate the question quality, two sets of multiple linear regression models are applied – one to predict the question score and the second one to predict the number of answers. For each set, four different regression models are applied and compared in order to discover which independent variables have the most predictive power. Each set of models uses the same dependent variable and a different set of independent variables. Table 6 gives an overview of a model set. Model 0 is a reduced intercept only model without any predictors and is used as a baseline model. Model 1 includes only the tags and the intercept. Model 2 contains only the question related features except the terms – title question length, body question length, code snippet and tags. This model aims to explore whether the question score and the number of answers depend alone on the features which describe the question, excluding external characteristics such as user reputation. Model 3 uses all independent variable except for the terms, and Model 4 includes all independent variables. The motivation for this choice lies in the aim to investigate to what extent structural/lexical features such as terms influence the number of answers and the evaluation of the question. To compare the performance of the models in each set, the R-squared, MSE and the Mean Absolute Error (MAE) are reported. R-squared is a statistic measure to evaluate how well the model fits the data. The MSE measures the performance of the model by the average of the squares of the model errors where the model errors are the differences between the actual and predicted values. The MAE is similar to the MSE, but instead of squared errors, the MAE uses absolute errors. The lower the MSE and MAE values, the better the model performs.

Table 6. A model set

	Independent variables
Model 0	intercept
Model 1	tags
Model 2	question title length, question body length, code snippet and tags
Model 3	question title length, question body length, code snippet, user reputation, tags
Model 4	question title length, question body length, code snippet, user reputation, tags and terms

In a second stage of the analysis, I investigate and analyze the terms with the highest predictive power based on the results of model 4. In the term analysis only those terms that have a statistically significant influence on the question score and the number of answers are included. As the total number of those terms is very large – 3259 terms predicting the question score and 3818 terms predicting the number of answers, only part of them were selected for the qualitative analysis. For this research I am interested in the terms which influence the question quality the most. The significant terms were therefore ranked based on their coefficient value. The terms with high coefficient values contribute to a high question score and number of answers and terms with low coefficient values determine questions with a low score and low number of answers. I extract the 10 percent of the terms with the highest and 10 percent of the terms with the lowest coefficient values. I assume that this percentage provides enough terms to discover patterns. The patterns are used to formulate recommendations to users as to which terms they should include in order to increase the question quality.

4. Results

This chapter presents the results of the multiple linear regression models. In section 4.a the results of the models with dependent variables *number of answers* and *question score* are reported. More precisely, the MSE and MAE of the models applied on the validation set and the test set are compared with each other and the coefficients of the best performing models are presented. Section 4.b discusses the outcomes of the second-stage analysis by indicating which terms have the greatest influence on question quality.

a. Ridge regression results for question score and number of answers

As MSE and MAE are measures of how close the predicted values are to the real values, it is desirable to achieve MSE and MAE for the set similar to the values from the training set. If the values do not deviate too much, it can be concluded that the built model is robust and has strong predictive power. To find the optimal parameter, the models were trained with different α values in increasing order on the training set, tried on the validation set and the α that gives the lowest MSE on the validation set is chosen as the optimal parameter and applied on the test set to evaluate the predictive validity of the models. Model 1, 2 and 3 for predicting question score and number of answers were found to have the lowest MSE, MSE = 2.667 and MSE = 1.198, respectively, when the alpha parameter is 20. Model 4 has a much higher parameter of 400 for both dependent variables.

The results of the models for predicting question score and number of answers are presented in Table 7 and Table 8, and Figure 1 and Figure 2. It is striking that for all models, the MSE and MAE scores of the test set are smaller than the same values of the training set. That would mean that the models can predict more accurate on new datasets than on the data they were built on which is somewhat surprising given the size of the datasets. One possible explanation could be the difference between the mean values of the question score and the number of answers in the training and the test set: $M_{qs\ training}$ = 1.424, $M_{qs\ test}$ = 1.226, $M_{na\ training}$ = 2.374, $M_{na\ test}$ = 2.074. One possible reason for the difference in the mean values is the fact that the first 60% of the questions were used as training set, the next 20% as validation set and another 20% for the test set. As the questions are in chronological order, it could be the case that the question in the test set had less time to be answered or voted on, thus having lower question scores and number of answers.

Table. 7 Results Ridge regression *question score*

	MSE	*MAE*	R^2	**F-statistic**
Model 0 training	6.177	1.538		
Model 0 validation	5.756	1.485		
Model 0 test	5.675	1.482		
Model 1 training	5.507	1.429	0.108	14.450
Model 1 validation	5.248	1.367	0.076	3.219
Model 1 test	5.138	1.375	0.088	3.768
Model 2 training	5.373	1.413	0.130	17.770
Model 2 validation	5.166	1.351	0.091	3.897
Model 2 test	5.063	1.363	0.102	4.396
Model 3 training	5.209	1.395	0.157	22.069
Model 3 validation	4.919	1.290	0.134	6.039
Model 3 test	4.869	1.323	0.136	6.124
Model 4 training	4.811	1.341	0.221	7.947
Model 4 validation	4.662	1.253	0.180	1.896
Model 4 test	4.622	1.286	0.180	1.897

Figure 1. Results MSE and MAE *question score*

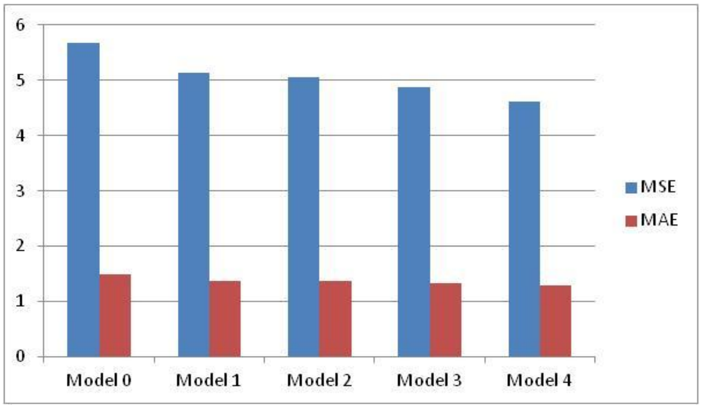
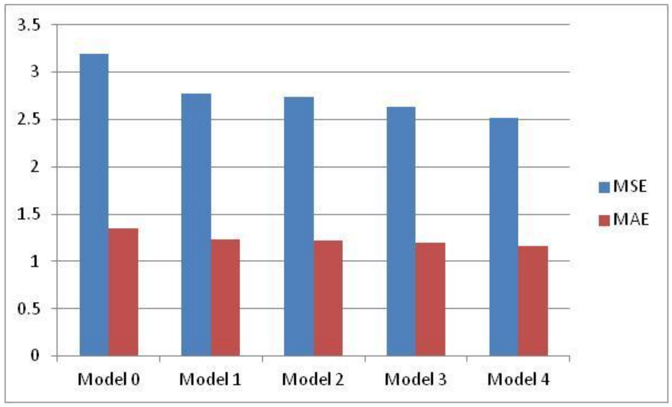


Table. 8 Results Ridge regression *number of answers*

	MSE	*MAE*	R^2	F-statistic
Model 0 training	3.757	1.403		
Model 0 validation	3.088	1.338		
Model 0 test	3.199	1.353		
Model 1 training	3.173	1.272	0.155	21.841
Model 1 validation	2.697	1.209	0.088	3.755
Model 1 test	2.769	1.228	0.109	4.771
Model 2 training	3.074	1.256	0.182	26.359
Model 2 validation	2.667	1.198	0.098	4.235
Model 2 test	2.738	1.219	0.119	5.257
Model 3 training	2.975	1.240	0.208	31.163
Model 3 validation	2.524	1.158	0.146	6.672
Model 3 test	2.630	1.192	0.154	7.079
Model 4 training	2.736	1.194	0.272	10.431
Model 4 validation	2.408	1.130	0.186	1.974
Model 4 test	2.514	1.163	0.191	2.048

Figure 2. Results MSE and MAE *number of answers*



As it will be explained later, the tags are beyond the scope of the current research and will not be discussed in detail. Instead, the overall influence of the tags on the prediction of question quality was investigated. From Table 7 and 8 it can be concluded that Model 1, including the tags, performs better than the baseline model for both, question score and number of answers, as the MSE has lower values - $MSE_{qs\ model\ 0\ test}$ = 5.675 and $MSE_{qs\ model\ 1\ test}$ = 5.138; $MSE_{na\ model\ 0\ test}$ = 3.199 and $MSE_{na\ model\ 1\ test}$ = 2.769. Compared to Model 2 however, where the question title and body length and the presence of a code snippet are included, the performance does not change drastically. The MSE of Model 2 for predicting question score decreases with only 0.075 and for number of answers with only 0.031. Also the increase of the R-squared values for both models is rather small - 0.014 for question score and 0.010 for number of answers. These results indicate that the tags do influence the question quality, whereas the inclusion of the title length, body length and the presence of a code snippet gives a minor improvement.

For both model sets it applies that the more complex the model is, the better it performs on the training and test set – MSE and MAE decrease with the increase of the number of independent variables and all models outperform the baseline Model 0. This implies that Model 4 for both question score and number of answers fits the data best. The same conclusion can be drawn from the R-squared values. For number of answers, the R-squared for the test set increases from 0.102 for Model 2 to 0.180 for Model 4, meaning that Model 2 explains 10.2% of the variance in the question score in the test set while Model 4 explains 18.0%. Similarly, with regard to number of answers, the R-squared values for Models 1 and 3 for the test set are 0.119 and 0.191, respectively.

Table 9 provides information about the regression coefficients and their significance for all independent variables, except for the tags and terms. The coefficients with the highest and lowest absolute values for terms are presented in Tables A5 and A6 in the appendix. I do not discuss the tags because of the large number. I consider the interpretation of these control variables beyond the scope of the current research. The terms are further explained in section 4b. As Model 4 is the model that has the best performance, only the coefficients of Model 4 are presented and discussed. From the table it can be seen that the question title and body length and the presence of a code snippet, have a significant negative effect on question score and number of answers, while reputation has a positive effect. The high degree of significance of the coefficients is not surprising and is due to the large number of observations. To better understand the effect size, I calculate the effect of a ten percent increase in body length, title length and user reputation, while taking the natural logarithm into account. A ten percent increase in title length, body length and user reputation results in a change in the questions score of -0.010, -0.019 and 0.015, respectively. Including a code snippet reduces the question score by -0.155. Hence, the effect of all variables is fairly small. In Model 4, for the dependent variable *number of answers* the title length effect is $\beta_{tl} = -0.058$, which implies that, taking the mean title length as baseline and accounting for the logarithmic transformation, a ten percent increase in title length results in a 0.006 reduction in the number of answers. Similarly, a ten percent increase in body length, $\beta_{bl} = -0.132$, and user reputation, $\beta_{ur} = 0.122$, gives an increase in the number of answers of -0.013 and 0.012 respectively. Including a code snippet reduces the expected number of answers by -0.050. Similar to Model 4 for predicting *question score*, the effects of the independent variables are fairly small.

Table 9. Model 4 Coefficients of independent variables for question score and number of answers

Feature	**Coefficient Question score**	**p-value Question score**	**Coefficient Nr. Answers**	**p-value Nr. Answers**
Title Length	-0.101	<0.01	-0.058	<0.01
Body Length	-0.202	<0.01	-0.132	<0.01
Code Snippet	-0.155	<0.01	-0.050	<0.01
User Rep.	0.153	<0.01	0.122	<0.01

b. Terms

In the term analysis only terms were included that have a statistically significant influence on the question score and the number of answers. As the total number of those terms is very large – 3259 terms predicting the question score and 3818 terms predicting the number of answers, only part of them were selected for the qualitative analysis. For this research I am interested in the terms which influence the question quality the most. I analyze only terms that have a statistically significant effect on the question quality. The significant terms were then ranked based on their coefficient value. The terms with high coefficient values contribute to high question score and number of answers and terms with low coefficient values determine questions with low score and low number of answers. I extract the 10 percent of the terms with the highest and 10 percent of the terms with the lowest coefficient values. I assume that this percentage provides enough terms to discover patterns.

i. Parts of Speech

Excessive use of (only) one part of speech might also have influence on the question quality. For example, too many verbs in a sentence can make it sound heavy and wordy (Weber, 2007) and therefore unpleasant to read. Therefore, in Table 10, I report the number of nouns, verbs and adjectives of the extracted terms. To count them I made use of the Natural Language Toolkit (NLTK), a suite for text processing libraries and programs for classification, tokenization, stemming, parsing and semantic reasoning[8]. Most of the terms that predict question score are nouns. This is not surprising as nouns are used most frequently in natural language. For each dependent variable, I test whether the parts of speech differ between high and low quality questions. I perform a Chi-square test to compare the counts of the different parts of speech in the two groups. With regard to number of answers, the counts of parts of speech differ significantly between high and low quality ($\chi^2 = 37.362$, $df = 3$, $p < 0.01$). Particularly, the percentage of nouns is higher in the groups of terms predicting low question quality. At the same time the percentage of used adjectives is higher for high question quality. As adjectives are

[8] www.nltk.org

words that have a descriptive character and are used to assign a noun a specific property, it may be concluded, that questions with a low number of answers are less descriptive and maybe do not explain the information need clearly enough. For question score, the counts of parts of speech do not significantly differ between the high and low quality groups ($\chi^2 = 1.190$, $df = 3$, $p = 0.755$).

Table 10. Parts of speech

TERMS Number of Answers		
top 10% high	n	percentage
nouns	211	57.18
adjectives	50	13.55
verbs	50	13.55
others	58	15.72
top 10% low		
nouns	240	65.04
adjectives	33	8.94
verbs	71	19.24
others	25	6.78

TERMS Question Score		
top 10% high	n	percentage
nouns	166	53.55
adjectives	35	11.29
verbs	74	23.87
others	35	11.29
top 10% low		
nouns	175	56.45
adjectives	31	9.54
verbs	70	22.15
others	34	10.77

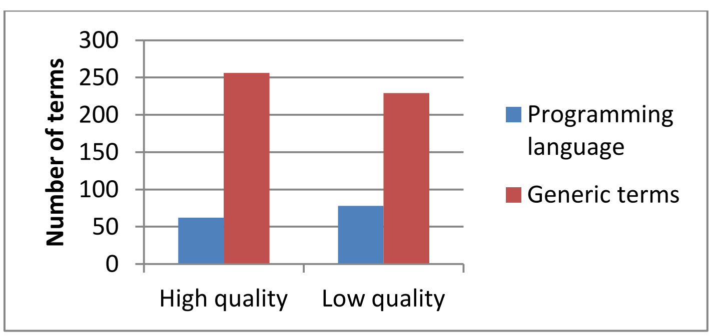
ii. Semantic analysis

The extracted terms are analyzed and first divided into two groups – professional/expertise terms and generic terms. In a second analysis only the generic terms will be considered and subdivided into several semantic groups. Often it is difficult to decide whether a term is a programming/expertise or generic term. To be able to make a distinction between the two groups, in the programming/expertise term set, I include strict programming/expertise terms such as *resig*, *dataframe*, and words that are considered expertise words, not commonly used in natural language conversation such as *deprecate*, *indention* etc. If it was doubtful which group a term belongs to, I used the Stack Overflow website for additional reference. The website was useful to recognize terms, such as *mythical* that refers to the Software Engineering book *The Mythical Man-Month* by Fred Brooks (1975), *girlfriend* that refers to the programming website

Cocoa is my Girlfriend[9] or *Floyd* referring to the Floyd-Warshall algorithm. These terms are considered professional/expertise terms as their references are not of common usage in daily conversations. The lists of terms are included in the Appendix. The proper nouns, like *Noel, Donald*, *Alexandrescu* etc., are mostly used as a reference and link to a new information source. As such references are too general and hard to interpret in the current research, they are considered proper names and therefore generic terms.

Tables A1, A2, A3 and A4 in the appendix give information about the selected terms. Table A5 and Table A6 in the appendix give an overview of all terms that were extracted to predict high and low question scores. Figures 3 and 4 show that, for both high and low quality questions, the generic terms dominate. For both dependent variables, I test whether there is a significant difference in the counts of generic and professional/expertise terms between high and low quality questions. Chi-square test indicate that the differences are significant: $\chi^2 = 6.833$, $df = 1$, $p < 0.01$ for question score and $\chi^2 = 24.189$, $df = 1$, $p < 0.01$ for number of answers. In the term group that contributes to low question score, the number of programming/expertise terms is larger. The same pattern is observed in the terms predicting the number of answers. To have a better understanding of the nature of the terms, a further distinction was made based on the semantic nature of the terms.

Figure 3. Question score



[9] http://www.cimgf.com/

Figure 4. Number of answers

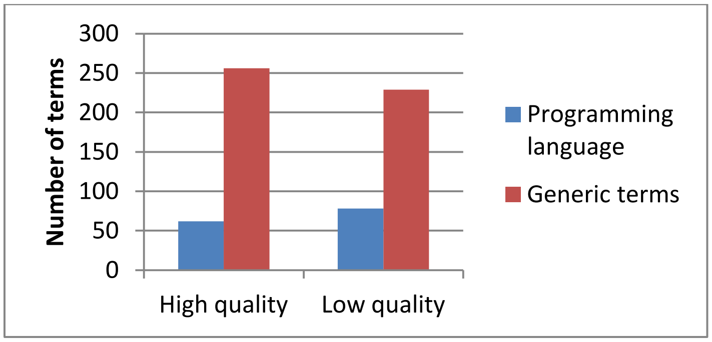


Table A3 and Table A4 shows the subgroups of the terms predicting the number of answers a question receives and Table A1 and Table A2 gives the terms that predict the question score. The terms predicting a high question score and a high number of answers, i.e. the terms with a positive influence on question quality, can be divided in subgroups where the following subgroups are very similar across the two dependent variables: Excitement, Negative experience/Frustration and Discussion/Explanations. The group of Excitement consists of terms which describe a passionate attitude towards a programming problem. These terms are assumed to be used by users who express excitement and emotional commitment to the subject in question. Terms of excitement that predict high question quality are *fascinated, compelling, praise, remarkably, aspiring, entertaining* etc. Similarly, terms such as *thrilled, believing, passion, amazed, favorite, enjoyed* account for a higher number of answers. The group of Negative experience/Frustration group consists of terms which express a negative emotion, mostly caused by lack of success when trying to solve a specific problem. When failing to find a desired solution people use negatively charged words such as *blatant, miserable, darned, disastrous, insanity, dread, sore* etc. which, according to the model results, indicate high question score. Examples of terms of negative experience or frustration that account for high number of answers are *horrific, miserable, torn, scare, evil etc.* Such high degree of frustration may be the results of multiple attempts to solve the problem which indicates that the user is providing a serious question. The third group lists terms that are used to start a discussions or explanations of a particular problem. Terms as *speculate, agree, disagree, advocate, argumentative* suggest an attempt to discussion, and *beware, misguided, unambiguous* assume that a user is trying to explain a specific issue. Although the words in this group seem related, they are less distinct and further research should perform a more in-depth analysis of this group.

In the list of terms that account for a high question score, two more subgroups were found: New members and Exceptions. The group of New members, including terms such as *newbies*, *newcomers*, *freshman* and *entrant* determines questions posted by, as the name already says, new members. Apparently, when users admit that they are new in the programming world, their question is appreciated by other new users or welcomed by experienced users who remember their first programming steps. The terms in the last group of Exceptions are used to discuss exceptional programming issues. In such questions users use words like *peculiarity*, *obscurity, surprises*, *counterintuitive, unintentional, nontrivial*, *contradicting, unintuitive*. These cases seem to be intriguing and challenging for the community and are therefore more likely to be appreciated and highly graded.

The terms that have a negative effect on the question score and the number of answers have one subgroup in common - the group of the misspelled words. In the group of terms predicting a low number of answers 8.31% is not spelled correctly. Examples of such terms are: *workin, acessing, specifc, undestand, calles, dyanmic, plateform, froma, wtih, konw, grammer, enviornment, istead*, *construtor, dereferences*, *feedbacks*. In the group for predict low question score the following examples were found: *plateform, langauge, softwares, feedbacks, intergration, fro, appropiate, charaters, litle*, *othe*, *unecessary, firs, sugest, suposed*, *shld, abut, maby, ether*. It can be assumed that questions containing typos are not considered professional and worthy for the community. Such questions may not be taken seriously and users may refuse to spend time giving an answer. In the latter group also off topic terms and interjections that express sounds normally used in daily conversations and more common in speaking than in writing, such as *hmmm, hay, aha*, were found. The off topic terms regard topics different from solving a programmer problem or information need. To the off topic group belong terms, such as *hiring, professionals, specialist, bosses, graduate, webdeveloper, bachelor*, are used mostly in questions related to people searching for or offering a job, students searching for answers to problems for their bachelor thesis. Such questions may be considered as off topic and not worthy to community users.

5. Discussion

In this chapter the results from the previous sections are discussed. Thereafter the outcomes are positioned against existing research.

The aim of this study was to investigate to what extent the discussed features influence the number of answers and the question score a question receives, and whether it is possible to predict these measures of question quality. To answer these questions, two sets of multiple linear regression models were estimated. The models were trained on 60% of the dataset and another independent 20% were used to optimize the regularization parameter for each model. Finally, the models based on the optimal regularization parameters, were used to predict within the test set to evaluate the predictive power of the models. The results from both sets of models showed that the inclusion of linguistic information improves the prediction accuracy of the models. A further analysis of the extracted terms shows that they can be classified in subgroups based on their semantic nature. First, it is interesting to see that certain groups of generic terms have greater impact on question quality. Second, questions that contain terms regarding newcomers, attempts at discussion or explanation of a specific problem or strong commitment to the problem described in the questions are more likely to receive a high question score and a large number of answers. Finally, the questions that are considered not worthy of a positive evaluation or receiving an answer are questions that include typos or that are found to be off topic.

Correa et al. (2014) and Saha et al. (2013) performed research on the features that describe low-quality questions on Stack Overflow. Both find that deleted questions are questions that are considered poor quality and off topic. Saha et al. (2013) investigated the tags assigned to such questions and found that homework and job-hunting belong to the tags in deleted questions. The current research supports these findings, as terms regarding these two topics were found to have a significant negative effect on both quality measures, question score and number of answers. Terms as *freelance, specialist, bosses, hiring, bachelor, coursework, undergrad, graduate* were found to predict a low question score and a low number of answers.

Another clear characteristic of low quality questions are misspellings and typos. Online social media sources are often characterized by not following common writing rules (Agichtein et al., 2008). Not taking them into account seems to be not appreciated and considered unprofessional. Therefore, misspellings and typos are considered low visual quality and they were found to predict a low question score and a low number of answers.

With regard to the terms predicting high quality questions, the results of the current research revealed more similarities. Nasehi et al. (2012) described question types and considered the following groups: debug/corrective, need to know, how-to-do-it, seeking different solution. Truede et al. (2011) distinguish similar groups – decision help, error, how-to, discrepancy, review. All of these questions can be seen as seeking an explanation. To present their information need, askers use terms like *speculate*, *agree*, *disagree, unbelievable*, *argues* which were found to have a significant positive effect on the question quality.

Existing literature does not provide a consistent explanation of whether a code snippet increases the question score or the number of answers. In this research it was shown that the effect of the presence of a code snippet is negative which is in line with the statement of Asaduzzaman et al. (2013) who explained that the presence of a code snippet may have a negative effect on the number of answers if the code is hard to follow or if other users cannot see the problem.

There also is disagreement in previous work about the influence of the question title and question body length. Where some researchers stated that very short and very long question are more likely to obtain an answer (Yang et al., 2011), other work found that too short questions may miss important information and may therefore remain unanswered (Asaduzzaman et al., 2013). Correa et al. (2014) found that deleted questions have a higher number of characters in the question body than closed questions. The results of this study indicate that both the length of the question title and the body length negatively affect question quality. The current results thus are mostly in line with the findings of Correa et al. (2014). Although, title length, body length and the inclusion of a code snippet all have significant negative effects on question score and the number of answers, it must be noted, that all effects are rather small.

Regarding the independent variable user reputation, my results are in line with previous work. As Yang et al. (2011) also showed, users with a high reputation are more likely to receive an answer than new users who logically have a lower reputation. For both quality features, question score and number of answers, it was found that the higher the reputation, the higher the value of the quality measure.

6. Conclusion, implications and future research

This chapter provides the conclusion based on the research questions in Chapter 1. Also, the implications of the current research and future research are described. The following research questions were stated:

1. Which question features presumably influence the probability of obtaining an answer, the number of answers, and the question score?
2. To what extent do these question features influence the number of answers and the question score?
3. Based on these features, can we make predictions about the number of answers a question will receive and the question score it will obtain from the community users?

The first research question aimed to give an overview on the question features which according to the literature may influence the question quality. For this purpose only features were discussed which are available at the time the question is posted - title and body length, the presence of a code snippet, user reputation and tags and terms used in the question. The features were presented and explained on the basis of previous research.

The second research question aimed to investigate to what extent the above mentioned features have an influence on the question score and the number of answers a question receives. In contrast to some previous research (Treude et al., 2011; Asaduzzaman et al., 2013), I found that the presence of a code snippet has a negative effect on the question score and the number of answers. Future research should reveal whether the length of the code snippet has an influence on the number of answers and the question score. One could assume that shorter codes are easier to follow, resulting in more answers. On the other hand, questions containing a longer code could also receive a lot of answers as they are more complex and could have different solutions. Longer codes are also expected to be posted by more experienced users who herewith contribute to the expertise content of the Stack Overflow website. Such question could be appreciated more by the community users and therefore awarded a higher score. Future work should investigate whether the length of the code snippet matters in predicting question quality, and if so, in what direction.

More research should also be conducted on the title and body length. As Yang et al. (2011) suggested, the question length could have a non-linear effect on the probability of receiving an answer. To obtain more clarity about the influence of the title and body length, the length could be divided in subcategories, e.g. 10 categories ranging from small to large. By comparing the parameter estimates for the different subcategories one could get a better understanding of the influence of the title and body length on the question quality.

The results showed that user reputation has a significant positive effect on the question quality – the higher the asker's reputation, the higher the question score and number of answers. However, the reputation on Stack Overflow is partly built on the votes a user receives when she posts a question. The questions' votes of a user that has asked less questions would be therefore negatively influenced by the reputation she has already obtained. From the current research it is not clear whether a question posted by a user with a low reputation receives a low vote because of the low quality of the questions or because the community users were influenced by the asker's reputation when deciding whether to answer or upvote the question. Future empirical research should show whether it is the user reputation that influences the question score and the number of answers or whether they are simply related variables.

The third research question aimed to investigate whether predictions can be made about the number of answers and the question score a question will obtain. The results from the test set follow those obtained for the training set: all models perform better than the baseline model and the more complex the model is, the better predictive power it has. Importantly, for both the training set and the test set, the model including terms clearly outperforms the baseline and all other benchmark models. For both, question score and number of answers, terms regarding excitement, negative experience/frustration and discussion/explanations account for high question quality. However, a more interesting and striking subgroup that both dependent variables have in common, is the group of the misspelled words that accounts for bad quality questions. It appears that questions containing spelling errors receive a lower score and number of answers. However, the current research does not provide information about the content of these particular questions and therefore does not make clear whether the removal of the spelling error(s) would improve the question quality. More specifically, if a user posts a question containing spelling errors that is inappropriate for Stack Overflow, it might not be rewarded by the user community with a high score or number of answers even if it is well-formulated in

English. In other cases, where a user posts a question that contributes to the community but accidentally contains spelling errors, the removal of the error might improve the question quality. Future empirical research should thus investigate whether questions with spelling errors are inherently bad questions or whether they are good questions that the community merely perceives as having low quality because of the spelling errors. Results from such research will help to understand what effect, for example, a spelling checker could have.

Another interesting improvement of the current study regards the terms found to have influence on the question quality. In the current study an attempt was made to include lexical entities, the terms, to predict question quality above the level of the assigned tags. However, the terms were analyzed manually, based on human judgment. This is rather subjective and may result in a somewhat arbitrary assessment. An automated way to analyze the extracted terms would be an improvement and a good suggestion for future research.

The results of the current study could be used for the development of an application that suggests terms that could be used in a question to increase the attention of community members and consequently the chance of receiving a larger number of answers or a higher question score. The application could also give advice about more frequent usage of adjectives to obtain a larger number of answers. Also, new users should be advised to indicate that they are new in the community as it helps to receive a high number of answers and high question score. Users can also be warned when their question title or question body is too short or too long.

Further areas for future research include analyses of other, less professional Community Question Answering websites to verify whether terms used to formulate questions also play an important role on these sites. Finally, instead of focusing on single terms, future research could pay attention to combinations of terms or phrases.

References

Adamic, L. A., Zhang, J., Bakshy, E., & Ackerman, M. S. (2008). Knowledge sharing and yahoo answers: Everyone knows something. *Proceedings of the 17th international conference on the World Wide Web*

Agichtein, E., Castillo, C., Donato, D., Gionis, A., & Mishne, G. (2008). Finding high-quality content in social media. *Proceedings of WSDM*.

Anderson, C. (2006). *The long tail: Why the future of Business is selling less of more.* Hyperion

Anderson, A., Huttenlocher, D. P., Kleinberg, J. M., & Leskovec, J. (2012). Discovering value from community activity on focused question answering sites: a case study of Stack Overflow. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 850–858.

Asaduzzaman, M., Mashiyat, A. S., Roy, C. K. & Schneider, K. (2013) A. Answering Questions about Unanswered Questions of Stack Overflow. *Proceedings of Mining Software Repositories*, 97–100.

Blei, D., & McAuliffe, J. (2007). Supervised topic models. Proceedings of Neural Information Processing Systems, 121-128.

Brooks Jr., F.P. (1975). *The Mythical Man-Month*. Addison-Wesley

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3.

Chang, D., Schiff, M., & Wu, W. (2013). Eliciting Answers on StackOverflow. Working Paper

Correa, D., & Sureka, A. (2013). Fit or unfit: Analysis and prediction of ’closed questions’ on stack overflow. In: *Proceedings of the ACM Conference on Online Social Networks,* Boston, MA, USA.

Correa, D., & Sureka, A. (2014). Chaff from the Wheat: Categorization and Modeling of Deleted Question on StackOverflow. Working Paper.

Field, A. (2009). *Discovering statistics using SPSS*. London: Sage.

Hartmann A., Van der Kooij A.J., & Zeeck A. (2009). Exploring nonlinear relations: models of clinical decision making by regression with optimal scaling. *Psychother Res*, 19, 482–492.

Hoerl, A. E., & Kennard, R. (1970a). Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12, 69-82.

Hoerl, A. E., & Kennard, R. (1970b). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12, 55-67

Jeon, J., Croft, W.B., Lee, J.H., & Park, S. (2006). A framework to predict the quality of answers with non-textual features. *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval,* 228–235. ACM Press.

LaToza, T. D., & Myers, B. A.(2010). Hard-to-answer questions about code. *Proceedings of the 2nd Workshop on the Evaluation and Usability of Programming Languages and Tools*. 81-86.

Li, B. , Jin, T., Lyu, M. R., King, I., & Mak, B. (2012) Analyzing and predicting questionquality in community question answering services. *Proceedings of the international conference on the World Wide Web*, 775–782.

Lou, J. , Lim, K., Fang, Y., & Peng, Z. (2011). Drivers of knowledge contribution quality and quantity in online question and answering communities. *Proceedings of the 15th Pacific Conference on Information Systems.*

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., & Hartmann., B. (2011). Design lessons from the fastest q&a site in the west. *Proceedings of the 2011 annual conference on Human factors in computing systems*, 2857–2866. ACM Press.

Manning, Ch. D., Raghavan, P., & Schütze, H.(2008). *Introduction to Information Retrieval*. Cambridge University Press

Nasehi, S. M., Sillito, J., Maurer, F., & Burns, C. (2012). What Makes a Good Code Example? A Study of Programming Q&A in StackOverflow. *Proceedings of the 28th IEEE International Conference on Software Maintenance*, 25-34.

Pal, A., Harper, F. M., & Konstan, J. A. (2012). Exploring question selection bias to identify experts and potential experts in community question answering. *ACM Transactions on Information Systems,* 30.

Saha, R.K., Saha, A.K., & Perry, D.E. (2013). Toward Understanding the Causes of Unanswered

Questions in Software Information Sites: A Case Study of Stack Overflow. *Proceedings of the 9th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering,* 663-666.

Suryanto, M. A. , Lim, E. P., Sun, A., & Chiang, R. H. L. (2009). Quality-aware collaborative question answering: methods and evaluation. *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, 142-151.

Treude, C., Barzilay, O., & Storey, M. A. (2011). How do programmers ask and answer questions on the web? *Proceedings of the 33rd ACM/IEEE International Conference on Software Engineering,* 2, 804-807.

Van der Kooij, A. J., & Meulman, J. J. (2008). *Regularization with Ridge penalties, the Lasso, and the elastic net for regression with optimal scaling transformations.* Manuscript submitted for publication.

Wang, S., Lo, D., & Jiang, L. (2013). An empirical study on developer interaction in StackOverflow. *Proceedings of the 28th Annual ACM Symposium on Applied Computing.* 18-22.

Weber, W. (2007). Text visualization - what colors tell about a text. *Proceedings of the IEEE 11th International Conference on Information Visualization.* 354–362.

Welser, H. T., Gleave, E., Fisher, D., & Smith, M. (2007). Visualizing the signatures of social roles in online discussion groups. *Journal of Social Structure,* 8.

Yang, L. Bao, Lin, Q., Wu, X., Han, D. , Su, Z., & Yu, Y. (2011). Analyzing and predicting not-answered questions in community-based question answering services. *Proceedings, Association for the Advancement of Artificial Intelligence*. 1273–1278.

Appendix

Table A1 Question score high quality subgroups

DISCUSSION/ EXPLANATION	***FRUSTRATION***	***NEW MEMBERS***	***EXCITEMENT***	***EXCEPTIONS***
misconceptions	blatant	newcomers	fascinated	peculiarity
analogies	darned	newbies	compelling	obscurity
speculate	miserable	freshman	addicted	surprises
picturing	hassles	entrant	praise	counterintuitive
argues	disastrous		coolest	unintentional
contra	shocking		rigorous	nontrivial
advocate	rude		thoughtful	contradicting
disagree	insanity		entertaining	unintuitive
motivate	immature		advancing	
exploration	dread		remarkably	
adviced	sore		aspiring	
unbelievable	cringe		mentally	
debated	monstrosity		genuinely	
investigations	disguise		insightful	
motivations	foolishly		excuses	
argumentative	discourages		mastered	
worthwhile	brutal		religious	
unambiguous	incapable		devoted	
prescribed	lousy		enjoyed	
beware	ineffective		craziness	
encouraged	sucking		thrilled	
justifies	cracked		smarts	
encourages	scare		amazed	
misguided	torn		greeted	
encourages	stinks		downright	
discourages	evil		grossly	

argues	erroneously	beloved
mandated	intricacies	
disagree	harmful	
contra	horror	
justifies	nuisance	
	lapse	
	chaotic	
	cracks	

Table A2 Question score low quality subgroups

OFF TOPIC	***NONPROFI***	***SPELLING ERRORS***
hiring	hmmm	framwork
professionals	hay	plateform
specialist	aha	langauge
bosses		softwares
graduate		feedbacks
webdeveloper		intergration
curriculum		fro
bachelor		appropiate
		charaters
		litle
		othe
		unecessary
		firs
		sugest
		suposed
		shld
		abut
		maby
		ether

Table A3 Number of answers high quality subgroups

DISCUSSION/ EXPLANATION	***FRUSTRATION***	***EXCITEMENT***	***NEW MEMBERS***	***EXCEPTIONS***
advocates	pains	fascinated	newbies	obscurity
argues	doubtful	aspiring	amateur	incomprehensible
analogies	hinder	believer		unorthodox
misconceptions	unsolved	addicted		
justifies	rude	praise		
tradition	shocked	religious		
advocate	horrific	clearest		
equivalently	screams	remarkably		
knowlege	miserable	thoughtful		
clearify	torn	compelling		
defacto	brutal	flaming		
actaully	hardest	excuses		
reminding	incapable	coolest		
convince	beaten	downright		
argumentative	crippled	thrilled		
debated	ineffective	enjoyed		
yay	harmful	believing		
nay	scare	neatest		
infocenter	evil	genuinely		
encourages	painless	passion		
encouraged	annoyed	entertaining		
conception	dread	amazed		
viewpoint		pleasing		
disagree		favorite		
competent		believes		
discourage		peachy		
argued		recommanding		
educating		mentally		
stance		devoted		

worthwhile	smarts
reviewers	favors
corrects	preffered
clarify	prefers
motivate	

Table A4 Number of answers low quality terms
SPELLING ERRORS
workin
acessing
specifc
outter
undestand
enviornment
calles
plateform
necesary
retrived
unecessary
dyanmic
(new) comer
abut
istead
froma
wtih
dereferences
construtor
konw
shld
grammer
feedbacks
exisitng
currenlty
heterogenous
Gird

Table A5 Terms question score

QUESTION SCORE HIGH QUALITY	*COEFFICIENTS*	*QUESTION SCORE LOW QUALITY*	*COEFFICIENTS*
fascinated	2.740	whining	-1.354
addicted	2.289	applicability	-1.146
praise	2.232	practiced	-1.131
mentality	1.931	dreadful	-1.106
camps	1.900	delved	-1.093
rage	1.862	bites	-1.061
lippert	1.775	enthusiasm	-1.060
misconceptions	1.747	rightfully	-1.060
blatant	1.699	religiously	-1.052
contenders	1.684	berryl	-1.039
mandated	1.670	c0000005	-1.032
analogies	1.660	bachelor	-1.032
coolest	1.658	muddled	-1.021
speculate	1.584	creeping	-1.013
thoughtful	1.558	bore	-0.980
newcomers	1.544	intergration	-0.979
picturing	1.516	dabbled	-0.963
stackers	1.487	maby	-0.963
replays	1.417	combinatorics	-0.962
darned	1.408	pruned	-0.960
mythical	1.393	fancier	-0.960
mentally	1.381	velocities	-0.939
draining	1.351	dislikes	-0.938
mantra	1.344	bowels	-0.937
downright	1.343	honesty	-0.933
thedailywtf	1.312	abut	-0.921
mastered	1.299	imaginable	-0.884
miserable	1.291	maxim	-0.884
advancing	1.277	influential	-0.880
rigorous	1.272	secretary	-0.879

bounty	1.271	objectively	-0.873
typeclass	1.258	guns	-0.870
argues	1.246	mitigated	-0.869
kai	1.242	precisions	-0.866
excuses	1.242	dereferences	-0.861
gory	1.213	society	-0.857
duplications	1.205	puzzler	-0.853
hassles	1.188	wg21	-0.852
geoff	1.184	2pm	-0.838
fogcreek	1.181	sucked	-0.835
disastrous	1.156	automatize	-0.830
unboxed	1.154	chatty	-0.827
unintentional	1.144	uniques	-0.824
widespread	1.138	siva	-0.823
knife	1.134	rdms	-0.823
ridden	1.133	concentrated	-0.821
unleashed	1.131	appart	-0.820
genuinely	1.127	24th	-0.810
contra	1.124	parlance	-0.808
religious	1.123	legality	-0.806
obscurity	1.121	maintainers	-0.799
distracted	1.111	lisa	-0.796
favors	1.107	marvellous	-0.794
catchable	1.104	570	-0.789
100x	1.095	disregards	-0.788
deserves	1.091	feedbacks	-0.788
instinctively	1.089	circumvented	-0.782
flaming	1.089	caters	-0.780
shocking	1.088	solitaire	-0.779
romain	1.088	decorates	-0.777
spellings	1.084	drained	-0.768
daringfireball	1.080	demanded	-0.765
reimport	1.079	evils	-0.765
restructuredtext	1.073	rewarding	-0.763

rude	1.073	shld	-0.762
nowadays	1.063	webdeveloper	-0.760
drilled	1.060	unacceptably	-0.758
compelling	1.038	enjoyable	-0.758
hone	1.035	unavoidable	-0.755
slashdot	1.034	lifetimes	-0.753
noel	1.034	focal	-0.751
commitanimations	1.034	suposed	-0.749
insanity	1.032	nlv	-0.749
equivalently	1.025	unfair	-0.745
swoop	1.025	exhausting	-0.743
advocate	1.011	rhyme	-0.735
immature	1.009	jcp	-0.733
developerfusion	1.003	gordon	-0.729
mutability	1.003	bloating	-0.723
practicality	1.003	inefficiencies	-0.723
recovers	1.002	hopping	-0.722
doprivileged	1.001	sugest	-0.715
disagree	1.001	firs	-0.713
pricey	1.000	eluded	-0.708
dread	0.997	beings	-0.707
sore	0.992	maxed	-0.703
cringe	0.991	sketched	-0.700
donald	0.987	wander	-0.696
mathematician	0.985	kohan	-0.695
striving	0.983	abundance	-0.695
alexandrescu	0.983	housekeeping	-0.691
ienumerables	0.982	dubious	-0.690
496	0.981	archival	-0.690
monstrosity	0.972	geany	-0.686
influences	0.972	pictured	-0.685
incurs	0.971	18th	-0.683
motivate	0.969	contemporary	-0.682
exploration	0.968	accesscontroller	-0.681

contravariance	0.967	20s	-0.678
cerr	0.963	classs	-0.677
bloch	0.956	vs05	-0.675
crockford	0.956	unecessary	-0.675
unmaintained	0.955	imran	-0.674
disguise	0.954	leisure	-0.673
foolishly	0.954	infeasible	-0.669
frontends	0.945	stephenwalther	-0.665
deprecate	0.944	degenerate	-0.664
patents	0.939	tandem	-0.663
undertake	0.936	undertaking	-0.659
surprises	0.936	mueller	-0.659
adviced	0.935	unneccessary	-0.659
354	0.935	bizzare	-0.656
senses	0.934	organizes	-0.654
lone	0.930	guaranteeing	-0.648
entertaining	0.930	revolve	-0.648
nintendo	0.929	answerer	-0.648
__class__	0.925	grounds	-0.647
landed	0.921	clearest	-0.647
discourages	0.919	intricate	-0.645
insult	0.918	narrows	-0.644
jeffrey	0.917	gath	-0.642
nontrivial	0.915	ftping	-0.642
xkcd	0.913	nj	-0.641
unbelievable	0.911	anonymized	-0.638
debated	0.908	contravariant	-0.634
brutal	0.907	averse	-0.631
corey	0.903	debates	-0.629
investigations	0.897	javi	-0.628
application_authenticaterequest	0.894	uncover	-0.626
ocd	0.893	saurabh	-0.626
devoted	0.891	othe	-0.626
incapable	0.886	slips	-0.625

enjoyed	0.880	unconventional	-0.623
tucked	0.879	impatient	-0.622
yikes	0.877	liability	-0.620
craziness	0.877	identifiable	-0.619
6x	0.875	ruling	-0.619
bookkeeping	0.869	rooting	-0.617
degrading	0.868	unconditionally	-0.615
peculiarity	0.868	bubbled	-0.613
cracks	0.867	brains	-0.612
retract	0.866	capitalized	-0.612
stewart	0.862	467	-0.611
notebooks	0.862	indention	-0.611
overseas	0.860	customisable	-0.611
mindset	0.855	ignacio	-0.610
disruption	0.853	l10n	-0.608
atwood	0.851	nomenclature	-0.607
extremes	0.849	superficial	-0.606
dialects	0.845	600k	-0.603
lousy	0.844	obfuscators	-0.602
blogged	0.841	minimalist	-0.599
guido	0.841	hay	-0.598
covariant	0.840	supervisors	-0.598
thrilled	0.839	paolo	-0.596
digitalmars	0.839	undecided	-0.595
occurrs	0.838	humor	-0.595
entrant	0.837	fresher	-0.595
uidevice	0.837	steadily	-0.593
unambiguous	0.836	rebasing	-0.588
beefy	0.836	employs	-0.587
liners	0.835	ilogger	-0.586
purity	0.835	performclick	-0.586
howard	0.833	485	-0.584
ineffective	0.833	orderbydescending	-0.581
5fsystem	0.833	devguide	-0.580

constructions	0.832	flesler	-0.579
dipping	0.829	mere	-0.578
creep	0.826	stephan	-0.577
beloved	0.824	findout	-0.577
exacly	0.815	premises	-0.577
jared	0.815	parallelizing	-0.572
lego	0.814	toes	-0.572
remarkably	0.814	jean	-0.568
repercussions	0.814	synchronizes	-0.566
accumulating	0.812	whatwg	-0.566
retroactively	0.810	wikia	-0.565
third_edition	0.807	tossing	-0.565
disregarded	0.807	posed	-0.557
scratched	0.805	repost	-0.554
ancillary	0.805	enforcement	-0.554
yagni	0.801	reflectively	-0.552
beware	0.800	choise	-0.550
smarts	0.800	performancecounter	-0.549
resistance	0.800	concentrating	-0.548
yegge	0.799	digress	-0.548
indistinguishable	0.797	chemistry	-0.547
aspiring	0.797	decompilation	-0.547
conducting	0.796	professionals	-0.545
amazed	0.796	hurting	-0.543
overcoming	0.795	find_if	-0.542
gravell	0.793	disparate	-0.542
autocompleting	0.791	litle	-0.542
counterintuitive	0.790	charaters	-0.542
collectors	0.789	interoperable	-0.542
ful	0.788	era	-0.541
prescribed	0.785	robotics	-0.538
uiresponder	0.785	sdlc	-0.538
richer	0.784	datastructures	-0.537
significance	0.784	annoyances	-0.537

weaker	0.782	trawling	-0.536
sucking	0.781	bruteforce	-0.535
newbies	0.780	bosses	-0.535
bridges	0.779	variability	-0.535
announced	0.778	net1	-0.535
daughter	0.771	memorize	-0.532
encouraged	0.771	asserttrue	-0.529
educating	0.771	graduate	-0.529
monetary	0.767	bar2	-0.529
deadlines	0.767	hashlib	-0.527
adoption	0.767	availible	-0.525
twiddling	0.766	automata	-0.525
freshman	0.762	derivation	-0.522
implementors	0.760	unrecognised	-0.521
girlfriend	0.759	moot	-0.520
cracked	0.758	itanium	-0.520
swallow	0.756	earn	-0.519
hum	0.755	plateform	-0.517
emphasized	0.754	880	-0.516
superfluous	0.754	pitfall	-0.511
thereof	0.752	candy	-0.511
doctypes	0.751	opentk	-0.510
graphed	0.749	hardcore	-0.508
writings	0.748	aha	-0.508
upd2	0.747	retarded	-0.507
chosing	0.747	pole	-0.507
0xffff	0.746	outweigh	-0.506
justifies	0.743	graduation	-0.504
contradicting	0.742	integrations	-0.504
jonas	0.740	nettuts	-0.504
musical	0.740	fro	-0.502
scare	0.740	downvote	-0.499
grossly	0.738	strategic	-0.499
apaches	0.738	ambitious	-0.498

eclipses	0.738	paulh	-0.498
v11	0.730	fictional	-0.497
jprofiler	0.729	effectiveness	-0.497
currying	0.725	308	-0.494
inertia	0.725	510	-0.492
necessitates	0.722	mcu	-0.490
greeted	0.718	livejournal	-0.482
torn	0.718	handleevent	-0.480
exploits	0.715	scite	-0.478
stinks	0.714	specialist	-0.477
lacked	0.713	476	-0.477
unintuitive	0.713	21022	-0.476
motivations	0.712	framwork	-0.476
evil	0.712	sof	-0.475
wiktionary	0.709	categorization	-0.474
fieldb	0.708	stall	-0.474
setresult	0.707	expressing	-0.472
argumentative	0.707	nlp	-0.470
succesfull	0.705	enumerators	-0.469
whould	0.703	dictated	-0.468
3500	0.702	implying	-0.467
erroneously	0.701	hiring	-0.467
fileaccess	0.700	interdev	-0.462
compatiblity	0.700	brainstorming	-0.461
90s	0.699	revisited	-0.461
60fps	0.699	528	-0.460
andrea	0.698	divisors	-0.460
avoided	0.695	kiss	-0.458
coworker	0.694	christian	-0.458
insightful	0.693	malcolm	-0.457
autoincrementing	0.693	icontroller	-0.456
cheaply	0.692	370	-0.456
argued	0.692	hopeful	-0.451
inferring	0.690	ajaxian	-0.449

hamming	0.689	pragprog	-0.448
mandates	0.688	proud	-0.447
redownload	0.688	gbs	-0.447
480px	0.685	exclusions	-0.447
1980	0.684	striped	-0.446
aids	0.683	ether	-0.444
rationale	0.683	j	-0.441
exhaustion	0.683	gettingstarted	-0.441
14	0.682	pyobject	-0.439
worthwhile	0.681	1_46_1	-0.438
comeau	0.680	aspnetmvc	-0.437
zillion	0.679	hoops	-0.437
somesuch	0.678	seattle	-0.435
professionally	0.678	knowledgeable	-0.434
suboptimal	0.676	spoj	-0.433
theater	0.676	inverting	-0.431
580	0.675	dte	-0.430
intricacies	0.674	5fand	-0.429
harmful	0.673	revealing	-0.429
8	0.672	bonuses	-0.427
projet	0.672	lamda	-0.426
pythonic	0.671	develops	-0.426
cgcontextdrawimage	0.671	synopsis	-0.425
todd	0.671	orientated	-0.425
betweeen	0.670	wikibooks	-0.423
squeezed	0.670	cop	-0.422
coworkers	0.670	viewvc	-0.421
prelude	0.669	bugtracker	-0.421
sphinxsearch	0.669	knuth	-0.419
animationdidstop	0.668	softwares	-0.418
blindingly	0.668	selenic	-0.417
borderline	0.667	appropiate	-0.416
referenceequals	0.667	userscripts	-0.415
shareware	0.666	tablets	-0.414

urge	0.666	grass	-0.414
horror	0.666	tails	-0.413
nuisance	0.665	317	-0.413
397	0.664	defun	-0.412
alleviate	0.663	hmmm	-0.411
21st	0.662	cascaded	-0.411
younger	0.662	ditto	-0.408
precedes	0.662	vga	-0.408

Table A6. Terms number of answers

NUMBER OF ANSWERS HIGH QUALITY	*COEFFICIENTS*	*NUMBER OF ANSWERS LOW QUALITY*	*COEFFICIENTS*
pricey	1.391	hammering	-1.172
tolerable	1.349	foggy	-1.162
fascinated	1.345	corollary	-0.952
kernighan	1.232	rangana	-0.933
aspiring	1.208	subtracts	-0.922
eratosthenes	1.189	24th	-0.898
believer	1.157	sdlc	-0.897
addicted	1.149	imran	-0.893
contenders	1.146	rewarding	-0.888
advocates	1.126	construtor	-0.886
argues	1.055	applicability	-0.885
laughing	1.032	furqan	-0.869
praise	1.031	nlv	-0.861
religious	0.992	sriram	-0.848
corey	0.991	lenient	-0.844
sniffed	0.988	dereferences	-0.828
motivations	0.970	tastes	-0.821
analogies	0.947	standardhost	-0.810
techie	0.933	technic	-0.809
geeky	0.932	dislikes	-0.801
internationally	0.932	mercy	-0.799
misconceptions	0.931	ittai	-0.790
clearest	0.906	absorb	-0.770
practicality	0.905	monish	-0.760
justifies	0.886	vs05	-0.750
ifdefs	0.877	notions	-0.742
tradition	0.864	enrich	-0.729
mythical	0.862	esoteric	-0.727
painlessly	0.862	prioritized	-0.724
remarkably	0.845	focal	-0.708

657	0.840	rightfully	-0.708
thedailywtf	0.837	berryl	-0.707
thoughtful	0.836	iwant	-0.693
compelling	0.833	aql	-0.690
jbu	0.829	heterogenous	-0.684
resistance	0.825	yvan	-0.683
flaming	0.812	attracted	-0.680
dread	0.800	toes	-0.675
girlfriend	0.795	quantitative	-0.671
pieter	0.792	obfuscators	-0.670
resnum	0.772	enforcement	-0.669
excuses	0.768	aswan	-0.668
pains	0.763	redundantly	-0.664
owe	0.763	wtih	-0.660
avoidable	0.757	trusty	-0.655
1011	0.752	assemblysecurity	-0.653
developpers	0.752	elaborated	-0.635
bounty	0.752	delved	-0.632
stare	0.749	semesters	-0.627
replays	0.741	misha	-0.625
prohibitively	0.732	leisure	-0.624
bent	0.731	lisa	-0.621
assessing	0.725	decay	-0.618
untidy	0.721	sketched	-0.617
motivate	0.719	froma	-0.617
unintentional	0.717	vladimir	-0.614
insisting	0.717	coursework	-0.613
sprinkle	0.716	converge	-0.612
advocate	0.716	traversals	-0.608
obscurity	0.712	istead	-0.604
favors	0.710	pragprog	-0.603
equivalently	0.705	humongous	-0.598
knowlege	0.703	abut	-0.598
conjecture	0.703	unfair	-0.598

630	0.702	gordon	-0.597
distracted	0.692	onsite	-0.596
affordable	0.689	correlates	-0.596
coolest	0.689	juggling	-0.594
distraction	0.689	bruteforce	-0.593
compsci	0.686	forefront	-0.589
doubtful	0.685	assisting	-0.589
clearify	0.684	fresher	-0.589
creep	0.677	28android	-0.586
hinder	0.676	webcache	-0.584
downright	0.672	pekka	-0.582
update2	0.670	emerged	-0.581
invokeattr	0.669	comer	-0.581
8	0.667	uncover	-0.578
thrilled	0.665	circumvented	-0.577
httpchannelfactory	0.665	keyboardinterrupt	-0.573
avoidance	0.661	currenlty	-0.572
betweeen	0.660	recommendable	-0.571
5fthe	0.656	manners	-0.563
isnumeric	0.656	ironic	-0.563
advent	0.654	justifying	-0.563
touted	0.653	guns	-0.562
defacto	0.647	664	-0.560
coworkers	0.640	teja	-0.560
unsolved	0.637	payed	-0.559
coworker	0.637	revisited	-0.555
enjoyed	0.637	4m	-0.555
incomprehensible	0.634	researches	-0.554
516	0.629	dyanmic	-0.545
purposefully	0.628	ror3	-0.543
professionally	0.628	oftentimes	-0.542
usecases	0.628	paramount	-0.540
buck	0.628	moron	-0.540
buzzwords	0.627	collaborator	-0.540

rude	0.627	appeals	-0.535
michal	0.623	parlance	-0.533
shocked	0.621	honesty	-0.532
actaully	0.619	influencing	-0.532
reminding	0.619	shld	-0.531
100x	0.619	snags	-0.530
tbh	0.616	hazy	-0.527
horrific	0.613	overcomplicating	-0.526
ff2	0.612	swallows	-0.526
sharma	0.612	delicate	-0.522
sofware	0.611	0x80070002	-0.520
duties	0.611	demanded	-0.519
lone	0.611	maintainers	-0.519
immense	0.610	fart	-0.518
stake	0.610	deem	-0.518
screams	0.607	xt	-0.516
marius	0.607	unecessary	-0.516
camps	0.606	muddled	-0.515
convince	0.606	gird	-0.513
expressive	0.599	bookmarklets	-0.512
khz	0.599	society	-0.512
strncpy	0.598	exisitng	-0.511
unorthodox	0.597	485	-0.511
approximated	0.595	collaboratively	-0.511
buzzword	0.595	misalignment	-0.511
caracters	0.594	whim	-0.510
nay	0.594	mestika	-0.508
bjarne	0.589	flashed	-0.506
knife	0.587	364	-0.505
bucks	0.586	nomenclature	-0.504
developerfusion	0.585	bignum	-0.503
webdevelopment	0.583	ism	-0.503
reactions	0.582	compliment	-0.501
webbased	0.582	barf	-0.495

educating	0.576	attendees	-0.495
stricter	0.576	evidenced	-0.495
unreal	0.573	quasi	-0.495
forgiving	0.573	utilises	-0.491
textbooks	0.572	wee	-0.491
believing	0.570	java3d	-0.491
neatest	0.568	coast	-0.490
empirically	0.567	smt	-0.489
600mb	0.567	srs	-0.489
genuinely	0.567	secretary	-0.487
ddj	0.567	amending	-0.487
satisfactorily	0.566	bursts	-0.486
prudent	0.566	datefromstring	-0.486
restate	0.565	striking	-0.485
passion	0.564	initiative	-0.484
predecessors	0.564	arcade	-0.482
defensive	0.563	ved	-0.482
miserable	0.560	recurrent	-0.482
torn	0.557	veterans	-0.481
sessionfactoryimpl	0.557	retrived	-0.480
723	0.556	1g	-0.479
interweb	0.555	python3	-0.477
brutal	0.554	localise	-0.474
oft	0.553	lightly	-0.473
stance	0.552	neophyte	-0.473
gory	0.551	diverge	-0.473
ribbons	0.550	clinet	-0.472
standardizing	0.550	licences	-0.472
700mb	0.550	caters	-0.470
jakob	0.547	simple_html_dom	-0.470
6k	0.547	bachelor	-0.470
hardest	0.547	bbq	-0.469
tidier	0.546	wander	-0.468
mentally	0.546	necesary	-0.465

whould	0.544	talented	-0.465
2147467259	0.544	tooth	-0.464
decades	0.543	practiced	-0.463
retro	0.540	fruitful	-0.462
selves	0.539	turk	-0.462
incurs	0.537	merger	-0.461
sketching	0.537	plateform	-0.459
distinctly	0.537	calles	-0.458
mantra	0.537	satellites	-0.457
singleline	0.537	intricate	-0.457
cgcontextdrawimage	0.536	criticize	-0.456
wholesale	0.535	mnemonics	-0.456
worthwhile	0.534	pyobject	-0.456
nicholas	0.534	gpgpu	-0.454
ancillary	0.534	bidding	-0.453
dbas	0.533	cprofile	-0.453
myth	0.533	recalled	-0.451
entertaining	0.532	duncan	-0.451
incapable	0.532	stringbyevaluatingjavascrip tfromstring	-0.450
mysql_config	0.531	mason	-0.449
samuel	0.529	visualized	-0.448
argumentative	0.528	prototypal	-0.446
crippled	0.528	feedbacks	-0.445
archaic	0.526	dismay	-0.443
pleasing	0.526	reinsert	-0.443
progressed	0.525	call_user_func_array	-0.442
innovative	0.523	initialises	-0.441
nulling	0.522	intellisence	-0.441
prominent	0.521	antipattern	-0.441
landed	0.520	interned	-0.440
anecdotes	0.520	hashchange	-0.437
overboard	0.519	informal	-0.436
beaten	0.519	psychology	-0.435

debated	0.515	flatfile	-0.435
habit	0.514	templatized	-0.435
grunt	0.514	reconstructed	-0.435
feeble	0.514	rads	-0.435
extremes	0.513	grammer	-0.434
16k	0.511	howard	-0.434
browsershots	0.509	princeton	-0.433
styleguide	0.507	docwiki	-0.431
donwload	0.506	undergrad	-0.430
preffered	0.503	hopelessly	-0.429
unassigned	0.502	subpackages	-0.427
processexplorer	0.502	o0	-0.425
purity	0.502	enviornment	-0.424
herb	0.502	javavm	-0.424
trigonometric	0.501	undestand	-0.424
upd2	0.501	copyrighted	-0.422
65k	0.500	ramp	-0.422
ineffective	0.497	specialist	-0.421
versatile	0.497	spcontext	-0.419
bracketed	0.495	guido	-0.418
petzold	0.493	exponentiation	-0.418
newbies	0.492	outter	-0.418
grr	0.492	whitepapers	-0.417
equate	0.492	q4	-0.417
744	0.491	subarray	-0.416
harmful	0.490	ropes	-0.415
curlopt_postfields	0.490	webmin	-0.415
younger	0.488	insignificant	-0.414
measurable	0.488	customisation	-0.414
dataaccesslayer	0.487	moderators	-0.412
donald	0.487	ext4	-0.412
unwilling	0.487	carrot	-0.410
quantify	0.485	flesh	-0.410
stakeholders	0.485	rex	-0.409

politely	0.485	result2	-0.408
ienumberable	0.484	foremost	-0.408
favorite	0.484	instructors	-0.406
buildpath	0.484	flesler	-0.406
concentrate	0.483	robotics	-0.405
skinnable	0.478	0_24	-0.405
cnet	0.478	curriculum	-0.405
crossplatform	0.476	gfortran	-0.403
scare	0.476	hig	-0.401
richer	0.475	particulary	-0.400
squeezed	0.474	konw	-0.398
programmers	0.474	exhibited	-0.398
reversible	0.472	specifc	-0.396
jarring	0.472	whatsnew	-0.395
marcus	0.472	withevents	-0.394
engineered	0.470	techdoc	-0.393
abused	0.470	reed	-0.393
eclipses	0.469	dataobjects	-0.393
lossy	0.469	htmldoc	-0.391
bookmarked	0.469	hay	-0.391
843	0.469	disturbed	-0.391
354	0.468	automagic	-0.391
xkcd	0.468	lockup	-0.389
compatiblity	0.468	descend	-0.389
whomever	0.467	parental	-0.389
triplets	0.467	pbxproj	-0.388
smarts	0.466	webcrawler	-0.387
cy	0.466	shout	-0.387
mechanical	0.466	rrd	-0.386
yay	0.466	notations	-0.386
monetary	0.466	devtools	-0.385
reviewers	0.465	cf9	-0.384
368	0.464	dance	-0.384
7bit	0.464	thesaurus	-0.384

dreams	0.463	fraud	-0.384
evil	0.463	categorization	-0.383
5fthread	0.463	outsourced	-0.383
scramble	0.461	readerwriterlockslim	-0.382
entity_id	0.460	sieve	-0.381
infocenter	0.460	jr	-0.381
encourages	0.460	superclasses	-0.380
719	0.459	hypothesis	-0.380
striving	0.458	dissappear	-0.379
encouraged	0.457	analyst	-0.379
conception	0.457	flooding	-0.378
redownload	0.455	motivated	-0.378
viewpoint	0.453	workitems	-0.378
ankhsvn	0.452	chilkat	-0.376
workplace	0.451	decompiling	-0.376
announced	0.450	usualy	-0.375
lottery	0.450	class_eval	-0.374
tilted	0.449	wsf	-0.374
creativity	0.449	tacked	-0.373
90s	0.448	parametric	-0.372
amazed	0.446	acessing	-0.372
nologo	0.446	reliance	-0.372
disagree	0.445	21022	-0.370
stackoverflowers	0.445	javi	-0.370
upcasting	0.445	chi	-0.369
6gb	0.444	scalatest	-0.369
775	0.443	credited	-0.369
inferring	0.441	steganography	-0.369
resist	0.441	predictive	-0.368
c89	0.441	winmo	-0.368
1053	0.439	containstable	-0.368
prefers	0.439	commiteditingstyle	-0.368
ars	0.439	freelance	-0.366
typo3	0.437	wiser	-0.366

currentdirectory	0.436	adaptation	-0.366
widespread	0.436	64bits	-0.366
favourite	0.435	haskellwiki	-0.366
persuade	0.435	albahari	-0.365
deadlines	0.434	consent	-0.365
mindset	0.428	pearls	-0.365
tlds	0.427	reformatted	-0.365
5fbug	0.423	ildasm	-0.364
redeployed	0.423	enroll	-0.364
believes	0.423	rebasing	-0.363
boil	0.423	fuzz	-0.363
misinterpreting	0.422	editmode	-0.363
intricacies	0.421	mimics	-0.362
crank	0.421	476	-0.361
inconsistently	0.420	mockups	-0.360
amateur	0.420	workin	-0.360
interconnected	0.420	manpage	-0.359
transmits	0.420	knowledgeable	-0.358
devoted	0.419	typedescriptor	-0.356
320x240	0.419	socks5	-0.356
quaternions	0.418	inventing	-0.356
overseas	0.418	audacity	-0.355
1235	0.418	postbuild	-0.354
invalidation	0.416	impacting	-0.353
fubar	0.415	308	-0.353
president	0.414	reseting	-0.353
edit3	0.413	employs	-0.352
relevancy	0.412	stalled	-0.352
shareware	0.411	stripslashes	-0.352
strcat	0.411	emacswiki	-0.350
characteristic	0.411	bringtofront	-0.348
webapplications	0.411	commentary	-0.348
programmer	0.408	folded	-0.348
painless	0.407	ion	-0.348

blindingly	0.407	grabber	-0.348
peachy	0.407	electrical	-0.348
avoided	0.406	tradeoffs	-0.347
hvga	0.405	javaagent	-0.347
taste	0.404	microformats	-0.346
7890	0.401	locality	-0.345
32767	0.401	isequal	-0.345
oversimplified	0.401	drm	-0.345
wow	0.401	triplet	-0.344
4d	0.399	opendatabase	-0.344
pythonic	0.399	cooperate	-0.343
derivatives	0.399	lxr	-0.343
recommending	0.399	tmpl	-0.342
nowadays	0.398	filesystems	-0.342
nag	0.398	index1	-0.341
inexpensive	0.397	bnf	-0.341
liners	0.396	trained	-0.341
5flanguage	0.396	aging	-0.340
trashed	0.395	occupying	-0.340
consensus	0.394	methodbase	-0.339
personally	0.394	floyd	-0.339
5ms	0.393	dataoutputstream	-0.339
twiddling	0.392	interdev	-0.339
hasnext	0.390	sqlyog	-0.337
ankh	0.390	motors	-0.336
argued	0.389	161	-0.336
misconfigured	0.388	sweep	-0.335
discourage	0.387	370	-0.333
qualifies	0.386	augment	-0.333
hashcodes	0.385	massage	-0.332
museum	0.385	bay	-0.332
kudos	0.384	generalization	-0.332
mutability	0.384	372	-0.331
venture	0.383	mentor	-0.331