The Journal of Web Science, 2015, 1: 1–15

ACQUA: Automated Community-based Question Answering through the Discretisation of Shallow Linguistic Features

George Gkotsis[1], Maria Liakata[2], Carlos Pedrinaci[1], Karen Stepanyan[3] and John Domingue[1]

[1] *Knowledge Media Institute*
The Open University
Milton Keynes, UK
firstname.lastname@open.ac.uk

[2] *Department of Computer Science*
University of Warwick
Coventry, UK
m.liakata@warwick.ac.uk

[3] *London School of Business and Management*
London, UK
Karen.Stepanyan@lsbm.ac.uk

ABSTRACT

This paper addresses the problem of determining the best answer in Community-based Question Answering (CQA) websites by focussing on the content. In particular, we present a novel system, ACQUA (http://acqua.kmi.open.ac.uk), that can be installed onto the majority of browsers as a plugin. The service offers a seamless and accurate prediction of the answer to be accepted. Our system is based on a novel approach for processing answers in CQAs. Previous research on this topic relies on the exploitation of community feedback on the answers, which involves rating of either users (e.g., reputation) or answers (e.g. scores manually assigned to answers). We propose a new technique that leverages the content/textual features of answers in a novel way. Our approach delivers better results than related linguistics-based solutions and manages to match rating-based approaches. More specifically, the gain in performance is achieved by rendering the values of these features into a discretised form. We also show how our technique manages to deliver equally good results in real-time settings, as opposed to having to rely on information not always readily available, such as user ratings and answer scores. We ran an evaluation on 21 StackExchange websites covering around 4 million questions and more than 8 million answers. We obtain 84% average precision and 70% recall, which shows that our technique is robust, effective, and widely applicable.

Keywords: Community Question Answering, Social Media, Natural Language Processing

ISSN 2332-4031; DOI 10.1561/106.00000001

1 Introduction

The proliferation of Community-based Question Answering (CQA) websites and their corresponding data has drawn the attention of computer science researchers in different areas. One of the intriguing problems in CQA research is the automatic identification of the best answer, which is expected to bring several benefits. First of all, since several answers are provided for each question, the readers of these websites will be able to process the candidate answers more efficiently and mitigate the "information overload" phenomenon. Secondly, a mechanism that identifies high quality answers will increase awareness within the community and will help to put more effort into questions that remain poorly answered. For instance, in StackOverflow[1] alone, as of September 2013, we found that approximately 33% of the questions have yet to be marked as resolved (i.e., out of the 5 million, 1.7 million questions have no answer marked as "accepted"). More generally, the study of the characteristics of answers is expected to improve our understanding of information seeking activities and social media reception in general.

Typically, CQAs adopt a simple model where the discussion is centred around a question posted by a user with the corresponding answers submitted by community members. A question remains "unresolved" until the questioner marks exactly one of the answers as the "accepted" one. Research so far has indicated that communities cannot be examined statically. In particular, the dynamic nature of on-line communication and communities alters the distribution of different roles in a community and may affect its sustainability (Rowe *et al.*, 2013). In this work we also discuss how the content/linguistic features differ between communities, how these features change over time and the implications this change has for the community's perception of good content quality.

[1] http://stackoverflow.com/

The study of publicly available corpora and the continuously increasing volume of user-generated content through social media is at the focus of web science. Researchers in related fields have used lexical, syntactic, and discourse features to produce a predictive model of readers' judgments (Pitler and Nenkova, 2008). In several cases, the use of shallow features, i.e. features that do not employ semantic or syntactic parsing such as sentence length (Feng *et al.*, 2010) or word length (Piantadosi *et al.*, 2011), have been shown to be effective in assessing properties such as ease of reading or usefulness. However, with respect to CQA, research efforts towards the exploitation of shallow features report relatively low results (e.g., Burel *et al.*, 2012 report 70% precision and Tian *et al.*, 2013 report 71% prediction accuracy for a balanced dataset). To improve the efficacy of their models, researchers refer to more contextual information, such as the *score* of each answer, the *comments* received or the *reputation* of the user.

Solutions that are based on *answer* or *user ratings* have been shown to be far more effective compared to linguistic ones. For instance Burel *et al.* (2012) achieve 85% precision largely due to the received score (answer rating), while Anderson *et al.* (2012) find that authors with a high reputation are behind good quality answers (user rating). At the same time, there is growing research interest around sites like StackExchange which employ badges and how this may affect the development of a community and the acceptance of answers. There is particular interest in studying well known behaviours, such as preferential attachment (the "rich get richer" effect), which may be a side-effect of systems that support community-based content assessment (Jones and Altadonna, 2012). In such cases, preferential attachment poses a threat to the development of the community, since the reputation framework reinforces the pre-existing community hierarchy.

In addition to the above concerns around the utilisation of reputation-based platforms, another issue pertains to the usage of answers' ratings, since these cannot be applied in a real-time setting due to the inherent delay between the answerer's submission and the community feedback. To provide a solution that is applicable in a real-time setting, we address the problem of best-answer identification in CQAs by leveraging purely textual features of the candidate answers. Our decision to ignore further contextual information is based on the fact that, when examining a question and its candidate answers, we do not always have at our disposal information such as answer ratings or the reputation information (e.g. new communities and users).

The main goal of our work is to address the problem of best answer identification and prediction using solely textual features. To do so, we examine 21 of the most active StackExchange websites, including the most popular one, StackOverflow. We study the evolution of language characteristics over time and across different communities. We investigate the distinct properties of accepted answers and we devise a classification strategy to achieve this prediction efficiently. Our paper makes the following contributions:

- We introduce a novel way of exploiting various shallow textual features with state-of-the-art performance that outperforms previous linguistics-based solutions.
- We evaluate and validate the results of the proposed technique on 21 StackExchange (SE) websites. To our knowledge, the scope and diversity of this evaluation is the largest so far.
- We show how our solution is generically applicable without the use of training data from the target SE website.
- We present a novel system – ACQUA – that implements the proposed solution and which is offered both as a web application and a web service. We present some early results from its usage and discuss why the feedback we have received until now is very promising.

Our paper extends previous work (Gkotsis *et al.*, 2014). More specifically, we extend the aforementioned paper with 3 major contributions: a) we present a system that implements this methodology, b) we extend the discussion on related work and position ours at a higher detail, and c) we evaluate the efficiency of our methodology by introducing more evaluation techniques.

The remainder of the paper is organised as follows: Section 2 reviews related work. Section 3 presents information around StackExchange and the corresponding dataset that we used. Section 4 introduces the features that we used for addressing our problem, including the proposed, novel methodology for devising discretised linguistic features. We then proceed to Section 5 where we present the results of our evaluation. Section 6 presents the ACQUA architecture and the corresponding web application. Finally, Section 7 discusses how our approach compares to others and makes a few general remarks about the task of best answer prediction as addressed here.

2 Related Work

The past years have seen the publication of several papers addressing the quality of answers in CQA. We first discuss work on best answer identification for StackExchange (SE) and Yahoo! Answers[2] (YA) and then move on to work pertaining to quality assessment of answers and textual content.

2.1 Answer quality prediction in CQA

The most recent work on best answer prediction in SE comes from Burel *et al.* (2012). The authors introduce three different classes of features for predicting the best answers. These classes contain features involving the content, user and thread information of answers. The combination of these features yields a precision of 85% and F-Measure of 0.84 for the case of two StackExchange websites (Server Fault and Cooking). Their evaluation shows that the success of the model deployed is mostly based on the "Score Ratio" feature (the proportion of scores given to an answer from all the scores received in a question thread). However, this feature constitutes part of "future knowledge", as the score value cannot be collected near the submission time of an answer. Moreover, when using purely

[2] http://answers.yahoo.com/

textual features, the authors report a precision drop for Server Fault[3] down to 65% and F-Measure down to 0.63. In our work we only consider textual content features, which are accessible immediately upon submission of an answer and we show how these features can be leveraged to obtain state-of-the-art performance.

Tian *et al.* (2013) share similar objectives with our work as they focus solely on the content of posts rather than user background information (e.g. user rating). They identify contextual information as the most important factor for successfully predicting the best answer. More specifically, they develop their model by using the questions together with the corresponding answers. However, some of the attributes used include comments, which are disregarded in our approach as they constitute future knowledge. This requirement for the existence of information such as comments is the reason why the dataset they used included only around 196k answers from StackOverflow, which were at least a year old. The final prediction accuracy reported in this case was 72%. Our solution overcomes this limitation for the need for long-lived questions and answers, and exhibits higher performance.

In general, YA adopts a similar operation mechanism to SE but differs in the nature of questions submitted by users, since questions are more debatable, subjective and are hosted on a single website divided into different thematic categories. Shah and Pomerantz (2010) construct a dataset of resolved questions each one containing exactly 5 answers (the ratio of answers is 4:1). They train a classifier using a number of shallow textual features, such as the length of the subject and content for each answer, as well as information about a user profile and the score received. The authors start by acknowledging that the baseline of the constructed dataset has an accuracy of 80% (i.e. negative classifier classifying all answers as non-accepted) and they manage to improve the classification accuracy up to 84.52%. The authors also report a lower performance when employing readability annotations from Mechanical Turk[4], due to the inherent subjectivity of the assessments. This is an important finding that demonstrates the subjectivity and difficulty inherent in best answer identification. Finally, Adamic *et al.* (2008) also focus on YA and introduce a number of thread and content features. Looking at questions under the "Programming" category, they report a prediction accuracy of 72.9% using features such as thread length, user number of best answers and user number of replies.

2.2 *Other work on CQA*

Work more broadly related to ours pertaining to CQA includes studying the activity of questions in StackExchange, such as whether a question will receive any answer (Yang *et al.*, 2011), or whether questions have been answered sufficiently (Anderson *et al.*, 2012). Yang *et al.* (2011) use the question length as a linguistic feature, in addition to 6 more features pertaining to the asker's background, and they experiment with different classification algorithms. The highest reported F-Measure is 0.325. Anderson *et al.* (2012) use several features to assess the longevity of a question and highlight the importance of the number of answers, the sum of scores on answers to question, as well as the length of the highest-scoring answer. Liu *et al.* (2013) present a framework for estimating question difficulty, following a competition-based approach that models together the level of question difficulty with the level of user expertise. While there is no use of the linguistic content when estimating question difficulty, they show a strong correlation between the difference in questions' difficulty ranking and their respective word distribution.

Work in Liu *et al.* (2011) predicts the satisfaction of Web searchers with existing CQA answers, introducing the concept of searcher satisfaction and breaking it down to the sub-tasks of query clarity, question match and answer quality. The paper uses a number of features, including character and word counts, but also non-linguistic features, to train logistic regression models on annotations from Amazon Mechanical Turk. They showed an improvement of answer-search rankings over a google search baseline.

Finally, numerous papers have been published that focus on the assessment of user-generated content quality. Jeon *et al.* (2006) define answer quality in terms of non-textual features such as click counts and answerer's acceptance ratio, and correlate each of these against manually judged quality scores. The features representative of quality are then incorporated into maximum entropy models and their results show that they can improve performance in retrieval experiments. Agichtein *et al.* (2008) use human editors to train a classifier for high and low quality questions and answers in YA. They use different features including baseline linguistic features such as word n-grams and report 67% precision (0.805 AUC) for an unbalanced dataset comprised of a few thousand answers. Furthermore, their study reports that the length of an answer is a significant indicator of answer quality.

2.3 *Measures of textual quality*

Some of the work on answers and other user-generated content considers textual content in terms of traditional readability measures (Gunning, 1968) and basic linguistic features (e.g. numbers of syllables per word, n-grams and even Part-Of-Speech (POS) tags), as in Agichtein *et al.* (2008), while the majority relies on external evidence of quality, such as community approval. However, there is a significant body of literature (Collins-Thompson, 2014) which investigates new measures of textual quality since traditional readability measures have been proven unsuitable for web documents, short or noisy texts. Such work models readability by means of baseline features (e.g. number of characters per word), vocabulary features (unigram language model), syntactic features (such as number of noun and verb phrases), measures of lexical cohesion, entity coherence and textual coherence. This research has been shown to provide reliable assessment of quality in texts ranging from news articles (Pitler and Nenkova, 2008), academic publications (Louis, 2012), scientific journalism (Louis, 2012; Louis and Nenkova, 2013), automatic summaries (Vadlapudi and Katragadda, 2010; Louis, 2012; Louis and Nenkova, 2014), teaching material from the web (Tanaka-Ishii *et al.*, 2010) and student essays (Yannakoudakis and Briscoe, 2012).

[3] http://serverfault.com

[4] https://www.mturk.com/

However such measures of textual quality have not yet been exploited in research on social media and on-line fora, presumably due to the difficulty in accessing the discourse and syntactic structure. Some recent work by Tan *et al.* (2014) examines the effect of the wording or phrasing of a tweet on its popularity. They test pairs of topic and author controlled tweets that differed in more than just spacing to understand why one is re-tweeted more than the other in its pair, counteracting for number of followers and timing of the two tweets. They found that a combination of features representing informativeness, language model, retweet-POS associations and readability outperformed a classifier that is based on timing and number of followers of the tweeters by more than 10%. This illustrates the importance of looking more carefully at the quality of linguistic content in social media to understand propagation and popularity.

We make use of some of the shallow readability linguistic features proposed by Pitler and Nenkova (2008), namely their baseline and vocabulary features, which are easy to compute, to train classifiers on accepted answers in CQAs in order to predict best answers. As will be explained in more detail in Section 4, we obtain a boost in performance when, rather than using the feature values directly, we sort them and discretise them. In a similar approach, Tanaka, Tezuka and Terada (2010) sort texts by constructing a readability comparator that, given two texts, will give an assessment of which of the two is more readable. The construction of the comparator only requires data annotated with two reading levels (difficult and easy). A sorting algorithm is applied based on comparisons of two documents' feature vectors using a binary SVM each time and readability is represented as the ranking of a text within the global ordering. Rather than sorting the answers, we sort feature values for each of the features and present their ranking (discretised value) as input to a binary classifier for accepted answer vs non-accepted answer.

3 StackExchange Dataset

StackExchange (SE) is the engine that powers some of the most popular CQAs such as StackOverflow (SO), Mathematics and Server Fault. Webpages in SE consist of one question and an arbitrary number of answers submitted by users. As of February 2014, 115 SE websites are available, each focusing on one topic. Topics are diverse, ranging from programming, system and network administrating to cooking, scientific skepticism and English language. As indicated in the mission statement, SE "is all about getting answers, it's not a discussion forum, there's no chit-chat". In order to maintain the quality of both questions and answers, posts are curated by the members of the community and if a question or an answer is deemed to be inappropriate or irrelevant, the post is removed from the website. In addition to the above, the reputation system introduced incentivises users to receive accreditation from the community and create high quality content, which is rewarded through badges and extra rights (such as the right of content removal). The high quality of the content has lead SE's premier website, StackOverflow (SO), to grow vigorously and attract almost 3 million users in approximately 5 years[5]. In total, as of February 2014, SE websites host 4.8 million users, 8.3 million questions and 14.7 million answers.

Table 1: Overview of the StackExchange websites dataset. Columns refer to the number of accepted (A), non-accepted (NA) and total number of answers (Total).

SE Website	A	NA	Total
stackoverflow.com	3,375,817	3,795,276	7,171,093
apple$^{\text{se.com}}$	14,471	14,149	28,620
askubuntu.com	37,907	33,746	71,653
drupal$^{\text{se.com}}$	14,393	8,558	22,951
electronics$^{\text{se.com}}$	11,726	14,942	26,668
english$^{\text{se.com}}$	17,369	31,617	48,986
gamedev$^{\text{se.com}}$	9,866	11,106	20,972
gaming$^{\text{se.com}}$	24,019	20,457	44,476
gis$^{\text{se.com}}$	10,015	8,724	18,739
math$^{\text{se.com}}$	98,351	78,294	176,645
mathoverflow.net	21,447	23,660	45,107
meta.stackoverflow.com	27,682	26,060	53,742
physics$^{\text{se.com}}$	10,851	10,389	21,240
programmers$^{\text{se.com}}$	15,998	52,694	68,692
serverfault.com	82,315	89,833	172,148
skeptics$^{\text{se.com}}$	2,041	1,421	3,462
stats$^{\text{se.com}}$	9,360	7,297	16,657
superuser.com	89,251	91,247	180,498
tex$^{\text{se.com}}$	30,642	20,249	50,891
unix$^{\text{se.com}}$	16,283	16,155	32,438
wordpress$^{\text{se.com}}$	19,420	10,788	30,208
Total	3,939,224	4,366,662	8,305,886

$^{\text{se.com}}$ .stackexchange.com

The full content – except users' personal information – of SE is distributed under a Creative Commons licence. For our work, we downloaded the dump of September 2013[6]. In addition to SO, our focus is on 20 of the biggest SE websites (in terms of generated content size), all of which are written in *English language*. The total number of answers in our dataset is over 12 million and the number of questions is almost 7 million. For the purposes of the evaluation study, we excluded content created by users that had their account removed or deleted. Furthermore, for evaluating the performance of our model classifier, we only kept questions with an accepted answer[7]. The resulting dataset contains more than 8 million answers and almost 4 million questions (see Table 1 for an overview).

4 Features for Best Answer Prediction

In this section we present the features used for training and evaluating our classifier. We initially present some shallow text features and one simple vocabulary, lexical-based feature. We then proceed by showing how we propose to exploit our features more efficiently. In order to assess the performance of the proposed model more holistically, we have also added a number of features referring to the rating of answers and users.

[5] http://stackexchange.com/sites

[6] http://www.clearbits.net/torrents/2155-sept-2013. The SE dump is now available from the Internet Archive https://archive.org/details/stackexchange.

[7] From now on, we will use the terms "accepted" and "best" for answers interchangeably.

4.1 Linguistic features

The term “shallow features” refers to those used by traditional *readability* metrics (Feng *et al.*, 2010) which have been used for several decades. The original purpose of these metrics was to estimate the average number of years of education required for being able to read and understand written text. The measurements use “surface”, aggregated values of text properties, such as the average word length, the average number of words in sentences or the number of sentences in a paragraph. In addition to being simple to understand, these features are computationally cheap compared to other more language-sensitive and context-sensitive features. More specifically, readability metrics are defined through a formula (based on regression analysis) that returns the expected number of years of education. Our metrics originate from similar yet more recent approaches. More specifically, we adopt as our baseline the baseline features in Pitler and Nenkova (2008), employed in the context of modelling readability judgements for the Wall Street Journal corpus, in terms of how well the articles are written. These features are the *average number of characters per word, average number of words per sentence, number of words in the longest sentence and answer length* (in terms of number of characters).

In addition to the above, we also considered using simple vocabulary features. Vocabulary features, compared to syntactic or discourse features, are cheap in terms of deployment (language-agnostic) as well as cost (linear time and space) and have been proven useful for content assessment (Callan and Eskenazi, 2007; Pitler and Nenkova, 2008). Other studies have examined how the language of a community evolves and affects the language use of individual members. Danescu *et al.* (2013) assessed the evolution of lexical corpora within an online community and use this change to predict a member’s lifecycle. To this effect we used a probability-based vocabulary feature from Pitler and Nenkova (2008) that is constructed from a unigram language model, where the probability of an answer is defined as:

$$\prod_{w} P(w|M)^{C(w)}$$

$P(w|M)$ is the probability of word w according to a background corpus M, and $C(w)$ is the number of times w appears in the answer. In our case, the background corpus is built from the content of each SE website separately.

The log likelihood (noted as LL from now on) of an answer is then:

$$\sum_{w} C(w)log(P(w|M))$$

Finally, in order to avoid any bias in favour of short answers, we normalise LL by dividing it over the number of unique words in the answer. Hence, this feature measures the probability of the answer being close to the vocabulary used by the SE community: the closer this value is to 0, the closer the answer is to the “community vocabulary”.

Figure 1 shows the average feature values for the accepted answers together with the non-accepted ones of SO using a one-month window time frame[8]. As seen from the figure, the linguistic features manage to clearly differentiate the accepted from the non-accepted answers. More specifically, accepted answers tend to be longer, use a less common vocabulary, contain longer words, more words per sentence and the longest sentences are lengthier. Even though the above remarks look promising concerning best answer prediction, when training a binary classifier *prediction* remains weak (58% precision and 0.56 F-Measure on average for all SE websites). Since the results that we obtained for a classification based on shallow features are comparable to similar approaches (e.g. Burel *et al.*, 2012; Tian *et al.*, 2013), these results will constitute our baseline for evaluating the proposed solution.

A more thorough investigation towards the explanation of this poor performance leads us to identify two main issues. Firstly, as illustrated in Figure 1, the characteristics of language evolve over time; in most SE websites users follow a more eloquent language (perhaps because of the increasing complexity of questions, or because of what is considered good practice and is rewarded accordingly). For example, the SE website on English language shows that, around early 2012, the average length of accepted answers is lower than the average length of non-accepted answers one year later. Hence, even though there is a steady gap between the values of accepted and non-accepted answers, the rapid change in the *absolute values* of the adopted shallow features is responsible for the poor classification.

We experimented with using a sliding window and examining the features in a narrow time frame (e.g., one month, as used for Figure 1). However, the large inherent *diversity* of the posts persists together with a large variance in values. Since this is not visible in Figure 1, we discuss one example regarding pertaining to the length feature: the average length of answers in SO during September 2008 is 482 characters with a standard deviation of 544. More specifically, for the same time period, the shortest accepted answer is only 2 characters[9] whereas the longest is around 18,000 characters. This deviation is also discussed in a later section, where features are presented all together.

Finally, even if a well-performing classifier existed for a single SE website and we used the features proposed above, the same classifier would have very low performance on another SE website. Indeed, as the reader may have anticipated, the characteristics of accepted answers vary significantly across the SE websites. For instance the accepted answers in Superuser have overall average length of 577 characters, whereas the corresponding value for Skeptics SE is 2,154 characters. Figure 2 presents the distinct characteristics for SO in comparison to Superuser. In this figure, each SE website takes into account the accepted answers together with the non-accepted ones. More generally, our analysis shows that the linguistic features for each SE website are unique and can be captured neither through absolute values nor universally. So there is very much a domain/community effect unique to each SE website. However, as already stated, our paper aims at developing a best answer

[8] Similar behaviour is identified for all SE websites and is omitted due to space limitations.

[9] “No” is the best answer to the question “Is there any difference between “string” and ‘string’ in Python?” http://stackoverflow.com/questions/143714

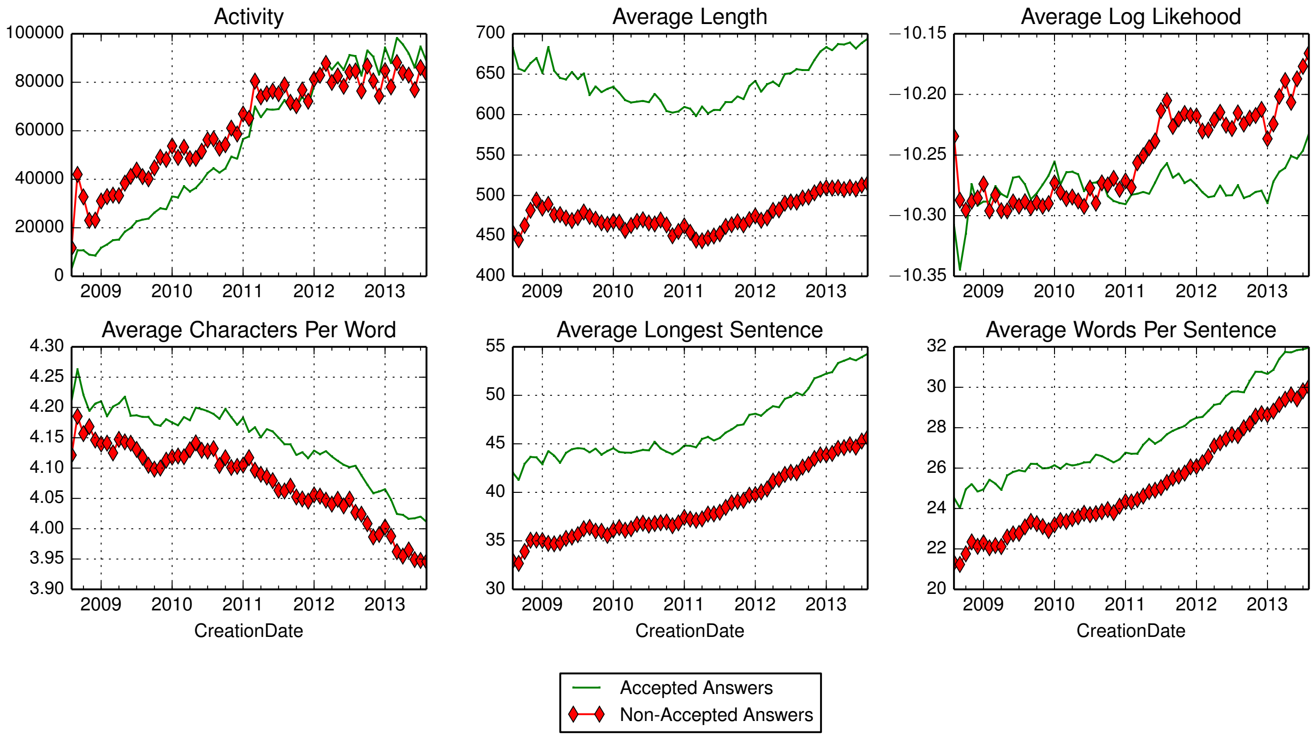


Figure 1: Activity and values of the linguistic features (y-axis) for the StackOverflow dataset over time (x-axis). Top left sub-plot shows the number of answers posted every month. The remaining sub-plots show the average values for the accepted and non-accepted answers.

prediction model independent of the community website.

4.2 Feature discretisation

In order to overcome the above weaknesses and effectively make use of the linguistic features introduced, our approach is to treat the collection of answers for *each question* as an *information unit* which can improve the training process. Instead of treating each answer independently of the other answers it is competing with, our approach is to assess the value of the features of each answer *in relation* to the corresponding features of its competitors. We introduce a new set of features that stem from the linguistic features used so far: instead of dealing with continuous values, these new features are the result of *grouping*, *sorting*, and *discretisation*.

We will present an example for the *Length* feature. Let us consider the example of Table 2, where for one question there are two candidate answers (i.e., question with Id 5 having answers with Id 6 and 7). We have already shown in Section 4.1 that the longer an answer is, the more likely it is to be accepted. In order to represent this preference, we group all answers by their corresponding questions (*grouping*). For each group, we then sort the answers (*sorting*) and assign a rank for each answer, starting from 1 and incrementing this rank by 1 (*discretisation*). Sorting is done either in descending or ascending order, so that the lowest rank is assigned to the answers that are marked as accepted (in this example, we use the information that longer answers are more likely to be accepted, hence descending order is conducted). For the example of Table 2, the answer with the longest *Length* will receive $Length_D$ of value 1 (answer Id 6 with length 250) while the answer that comes second a value of 2 (answer Id 7 with length 200 - note that we are representing the discretised form of each $feature$ as $feature_D$). The result of this process is the introduction of an equal number of linguistic features without the usage of any further information (apart from the necessary association of a question and its corresponding answers[10]).

Table 2: Example of feature discretisation for the case of *Length*, 5 submitted answers and 2 questions. Column Question Id refers to the question under which the answer is submitted.

Question Id	Answer Id	*Length*	$Length_D$
1	2	200	2
	3	150	3
	4	250	1
5	6	250	1
	7	200	2

As a result of the discretisation process on all of our shallow features, the information added and used for training purposes improved significantly. This is manifested by the information gain (about 20 times higher), which we present in the following subsection. Additionally, the benefits of this discretisation

[10] Note that other approaches typically omit this information.

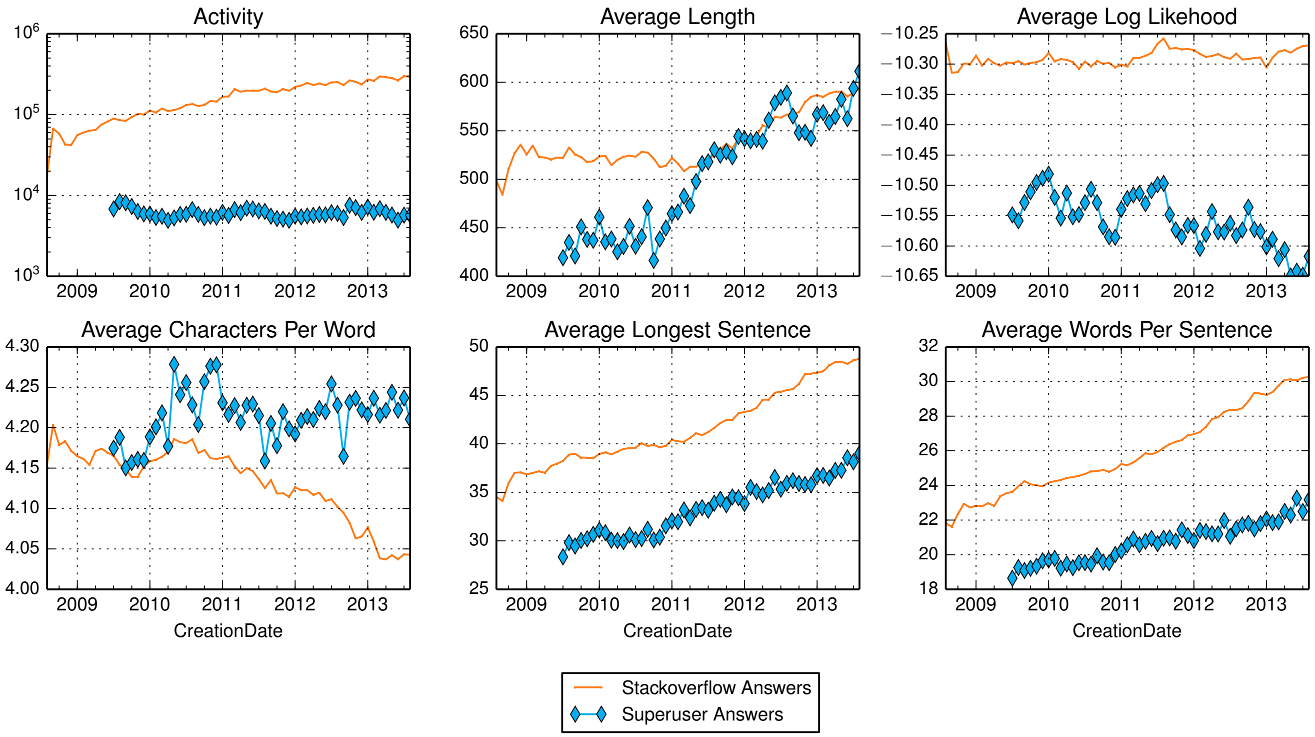


Figure 2: The differences in activity and linguistic features between StackOveflow and Superuser (accepted answers and non-accepted answers are presented together).

are discussed thoroughly in Section 5, where we present the classification results. It may appear that our discretisation process is dependent on "future knowledge", since discretised values may alter as more answers are submitted. Our method is no more time dependent than the notion of a best answer is, as it allows for best answer prediction at any point, in a real-time setting, which is not possible when relying on answer ratings. As more answers are entered, the discretised values change and a new current best answer can be derived.

In the following subsection we will discuss the inclusion in our classifier of two popular sets of non-linguistic features, to allow us a more thorough evaluation.

4.3 User and Answer Rating Features

Until now, we have discussed the linguistic features and how the proposed discretisation process is expected to yield better results. In order to have a more complete view of the performance of our classifier we have integrated some non-linguistic features. It is worth noting that these are included for evaluation purposes only; they do not form part of our approach. We group these features into different sets, following the discussion in Section 1. The first set of features (*user*) describes *past* or background knowledge and more specifically the *user profile*, such as the *reputation*, the number of *profile views*, number of *up-* and *down-votes* and the $UserUpDownVotes$ feature, which we define as the difference over the sum of *Up* and *Down* votes, as follows:

$$UserUpDownVotes = \frac{|UserUpVotes| - |UserDownVotes|}{|UserUpVotes| + |UserDownVotes|}$$

The second set of features (entitled as *Answer rating*), includes information concerning the community feedback on answers, such as the number of *comments*, the *score* and the *score ratio* ("the proportion of scores given to a post from all the scores received in a question thread", as indicated by Burel *et al.* (2012) as the most informative feature). Finally, another set of features (*Other*) was used, such as the *AnswerCount*, the *Age* (real number representing days) of answers and the corresponding $CreationDate_D$ (answer speed is linked to good answer quality, Anderson *et al.*, 2012). The total number of features is 21 and are shown in Table 3.

Table 3 shows the values for each feature in addition to their corresponding information gain. Notice that the first group of features (linguistic) as well as the *Age* are presented together with their *discretised* version for comparison purposes. The last column of the table is presenting the relative change in information gain when applying the discretisation process. Information gain is a measurement based on entropy used for machine learning and has been employed in classification tasks to identify important features. Information gain $InfoGain$ of an attribute A (e.g. $Length_D$) for class C (i.e. answer is accepted or non-accepted) is defined using the entropy H measurement as follows:

$$InfoGain(C, A) = H(C) - H(C|A)$$

Table 3: Summary of features used and their information gain. The last two columns indicate the information gain for the discretised features and the corresponding relative change due to discretisation. Values concern the averages for all SE websites.

Category	Name	Information Gain	Information Gain (Discretised Features)	Relative Change in Information Gain
Linguistic	*Length*	0.0226	0.2168	+8.59
	LongestSentence	0.0121	0.1750	+13.46
	LL	0.0053	0.1180	+21.26
	WordsPerSentence	0.0048	0.1404	+28.25
	CharactersPerWord	0.0052	0.1162	+21.35
Other	*Age*	0.0539	0.1575	+1.92
	AnswerCount	0.3270	-	-
User Rating	*UserReputation*	0.0836	-	-
	UserUpVotes	0.0535	-	-
	UserDownVotes	0.0412	-	-
	UserViews	0.0528	-	-
	UserUpDownVotes	0.0508	-	-
Answer rating	*Score*	0.0792	-	-
	CommentCount	0.0286	-	-
	ScoreRatio	0.4539	-	-

We can clearly see that the task of discretisation improves the information gain for all features. In particular, the information gain for *linguistic* features has increased on average 20 times. For the case of *Length*, the improvement is so significant that it manages to outperform well-known features, such as all those based on User Rating, and to rank as the third most important feature. At the same time, both $Length_D$ and $LongestSentence_D$ carry more information gain than $CreationDate_D$ which is also a popular feature shown to yield good performance.

Table 4: Results for best answer prediction using *linguistic* and *other* features with discretisation. Columns show macro averaged precision (P), recall (R), F-measure (FM) and Area-Under-Curve (AUC) using 10-fold validation.

SE Website	P	R	FM	AUC
stackoverflow.com	0.82	0.66	0.73	0.85
apple.stackexchange.com	0.84	0.68	0.75	0.86
askubuntu.com	0.84	0.74	0.79	0.88
drupal.stackexchange.com	0.87	0.79	0.83	0.89
electronics.stackexchange.com	0.79	0.65	0.71	0.84
english.stackexchange.com	0.77	0.52	0.62	0.83
gamedev.stackexchange.com	0.82	0.71	0.76	0.87
gaming.stackexchange.com	0.87	0.79	0.83	0.91
gis.stackexchange.com	0.85	0.73	0.78	0.87
math.stackexchange.com	0.85	0.74	0.79	0.87
mathoverflow.net	0.83	0.70	0.76	0.87
meta.stackoverflow.com	0.87	0.69	0.77	0.87
physics.stackexchange.com	0.86	0.71	0.78	0.88
programmers.stackexchange.com	0.76	0.40	0.52	0.84
serverfault.com	0.83	0.66	0.74	0.85
skeptics.stackexchange.com	0.87	0.83	0.85	0.91
stats.stackexchange.com	0.85	0.79	0.82	0.89
superuser.com	0.84	0.65	0.73	0.85
tex.stackexchange.com	0.87	0.77	0.82	0.88
unix.stackexchange.com	0.81	0.68	0.74	0.85
wordpress.stackexchange.com	0.88	0.80	0.84	0.89
Average	0.84	0.70	0.76	0.87

5 Evaluation: Best Answer Prediction

Having experimented with a number of different classifiers, our evaluation shows that we obtain the best results by using *Alternate Decision Trees* (ADT), (Freund and Mason, 1999). Even though we received good results with different classifiers available in Weka (Hall *et al.*, 2009), we attribute the high performance of ADTs to the fact that they constitute a well-known binary, boosting classifier for numerical data, which suits our goals. Our evaluation was conducted using 10-fold cross-validation. In order to verify the performance of the proposed solution we conducted different experiments, each one aiming at validating the characteristics of the proposed solution.

5.1 Prediction

Table 4 presents the first results concerning the performance of our classifier without the inclusion of features based on answer or user ratings. The table shows that the macro averaged (unweighted) precision using *linguistic* and *other* (namely *Age*, *AnswerCount* and the discretisation of Age, $CreationDate_D$) features with *discretisation* is *84%*. The remaining evaluation metrics (recall, F-Measure) maintain high values resulting in an average AUC of *0.87*. The website with the lowest precision is Programmers SE with 76%, which can be attributed to the fact that the dataset for this website is heavily imbalanced (only 23% of the dataset's answers are accepted – see Table 1). On the contrary, Skeptics SE has 87% precision with 0.91 AUC value, which can be explained as follows: Firstly 58% of the answers in the dataset are accepted (the third highest ratio from all SE websites). The second reason stems from the website topic and the type of discourse that takes place: questions in Skeptics SE mainly attract scientific reasoning without much technical information, hence prose and linguistic features play a more important role. This performance is also confirmed by the value of information gain for the discretised version of *Length*, which is 0.27 (Skeptics) whereas the average value for $Length_D$ is 0.22 (see Table 3). The English SE dataset is also imbalanced (only 35% of the answers are accepted, close to programmers SE), but language-based features manage to overcome this challenge, most likely due to the nature of the discourse (i.e. similar to skeptics SE). The resulting prediction has 77% precision and 0.83 AUC.

Table 5: Results for best answer prediction using different sets of features (Cases 1 to 6) for all SE websites. Columns show macro average precision (P), recall (R), F-Measure (FM) and Area-Under-Curve (AUC) for all 21 SE websites using 10-fold validation. Case 3 was presented in detail in Table 4.

No.	Features Used	P	R	FM	AUC
1	Linguistic	0.58	0.60	0.56	0.60
2	Linguistic & Discretisation	0.81	0.70	0.74	0.84
3	Linguistic & Discretisation & Other	0.84	0.70	0.76	0.87
4	Linguistic & Other & User Rating (no discretisation)	0.82	0.69	0.75	0.86
5	Linguistic & Other & User Rating (with discretisation)	0.82	0.72	0.77	0.88
6	All features (Answer and User Rating with discretisation)	0.88	0.85	0.86	0.94

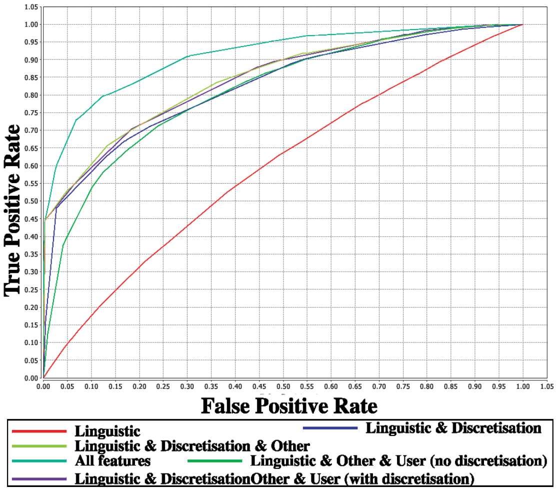


Figure 3: AUC for StackOverflow. Different curves show the results for 10-fold cross validation using different sets of features (Cases 1 to 6). The 4 overlapping curves in the middle show that the discretisation of features outperforms the linguistic-based approach (bottom curve), matches the classification based on reputation and approaches the classification using all features (top curve) including user and answer ratings.

5.2 *Improvement due to discretisation*

We have already shown the improvement in information gain after discretising the linguistic features (see Table 3). Here we aim to analyse the benefits of this process in the task of best answer prediction. To do so, we compare the performance of our classifier to other classifiers that use more sets of features, including features produced from ratings. Our goal in performing this comparison is to examine the information loss when choosing to disregard information coming from ratings.

Table 5 presents the results when using different sets of features and 10-fold validation. The table contains the average values for all SE websites as the output of different evaluations. Initially, we use the absolute values of textual features (also mentioned in Section 4) with low results (58% precision, Case 1). The second and third Cases both utilise the discretised features, while the third is additionally using the *other* set of features. Cases 2 and 3 constitute our proposed prediction method (Case 3 was presented in detail in subsection 5.1 and Table 4). Furthermore Case 4 refers to a "traditional" approach that relies in plain linguistics *and* user ratings. We can see that while a whole new set of features is added into the dataset, the performance of classification remains lower than Case 3, which is linguistics-based. Case 5 keeps the user ratings in addition to incorporating all features of Case 3. Hence, classification accuracy is the highest compared to all previous classifications, but almost identical to Case 3 which is strictly based on content and discretisation (higher F-Measure 0.77 vs. 0.76, higher AUC 0.88 vs. 0.87). Finally, Case 6 uses all features presented in Table 3, including the *answer ratings*. This set of features uses all features but most importantly user-entered scores and manages to outperform all of the previous cases. Case 6 shows that the information contained within answer ratings is independent – to a certain extent – of the information found in previous features.

In summary, results in Table 5 show that the discretisation of linguistic features manages to outperform significantly the classifier based on linguistic features only. Moreover, we can also see that user rating features such as reputation do not improve our classification, a sign that discretisation is a process that extracts very useful information and delivers very strong results. Figure 3 shows the AUC curves for StackOverflow for all 6 cases and confirms the above remarks.

5.3 *Generality*

The final part of our evaluation aims to examine whether our solution is generic enough to be applied without the need to train our classifier on data from a new website. If the answer to this question is positive, we can assume that our classifier is generic enough to be applied to almost any SE website and to a large extent contains cross-domain intuitions about the mechanics of best answer identification. In order to have a positive answer to our research question, two requirements must be satisfied. Firstly, our classifier should be able to describe the characteristics of the best answers accurately for each SE website (robustness). Secondly, the features used in this model must neutralise the special characteristics of each SE website (generality). To examine the above hypothesis, we created new datasets following a leave-one-out strategy for each SE website. For instance, for the case of English language SE, we merge the remaining 19 SE websites[11] into one training dataset and use English language as the test dataset. For the evaluation purposes we applied classification using the features of Case 3.

The results of the evaluation shows that the average values for our evaluation metrics remain intact. More specifically, average precision fell by 1%, while recall, F-Measure and AUC remained the same (see Case 3, Table 4 for the values). Hence, we can claim that our classifier manages to remain effective without requiring access to the specific knowledge of the SE

[11] We chose to exclude StackOverflow from training due to its large size which would slow the training process dramatically.

website. We believe that this result strengthens the value of discretisation even further. Despite the inherent variance in shallow feature values across answers and – even more – across SE websites, the discretisation process is able to demonstrate both robustness and generality.

Following the results of the above evaluation, we conducted a more detailed evaluation on generality. Figure 4 shows the heatmap of AUC *paired testing* between all SE websites. In our case, paired testing means we trained a classifier for each one of the SE datasets and tested it against all other 20 SE datasets. This process results in the evaluation of a recordset of roughly 3.5 billions. The results are illustrated as a squared *heatmap* of 21 × 21 dimension. Rows represent the SE website used for training and columns represent the SE website against which our model is being tested. The bar on the right of the heatmap is a legend that illustrates the colouring scheme applied. A cell coloured white means the maximum performance is achieved for the website being tested. As the colour of cells become more dark, the distance from the maximum AUC performance increases. When a cell is black, the maximum overall loss is reached (0.07 in our case). Hence by definition at least one white and one black cell is expected somewhere in the heatmap. Since it would be biased to test against the training dataset, we have replaced the left-to-right diagonal values with the 10-fold validation values found in previous generality evaluation subsection (i.e. values from leave-one-out strategy).

The average value (i.e. mean loss) for all cells in the heatmap is 0.009; this value is in agreement with our results from the Leave-One-Out evaluation. Meta SE is the worst for training returning 0.02 loss on average (the max loss is 0.07 for SO). The left-to-right diagonal is white which means that 10-fold validation has the best performance, as expected. A more close inspection on the values of the heatmap allows us to make two observations. Firstly, testing-wise, training for SO is the most challenging. More specifically, SO has average loss of 0.05, while Programmers is the best SE website (0.03 loss). Secondly, training-wise, using SO as a training dataset has the best overall performance: the average loss of 0.006; the highest loss is for English (0.02) and the lowest is 0.003 for both Serverfault and Superuser.

Overall, the primary objective of generality is achieved, since the mean loss is very low. This means that our approach is highly resilient in terms of the dataset used for training and the target community. Furthermore, the analysis of our evaluation results allows us to make two more observations: a) training with a large amount of data (SO) results in the best and most generic classifier, and b) training with small amount of data cannot be very effective when the target dataset is orders of magnitude bigger (SO).

6 ACQUA

ACQUA [http://acqua.kmi.open.ac.uk/] – which is an abbreviation for Automatic Community-based Question Answering – is an implementation of the approach proposed here and is available for all sites of the StackExchange Network[12]. ACQUA follows a server-client architecture for carrying out the task of best answer prediction. Users of the system can install the client on their browser and ACQUA provides visual indication of the answer predicted to be marked as the accepted one. Figure 5 shows a snippet of the web browser with ACQUA highlighting the best answer when visiting a specific SO question-webpage[13].

Our system is developed as a web application using the GreaseMonkey browser extension[14], available for all major browsers. The GreaseMonkey browser extension allows the installation and execution of JavaScript code for any website on the browser side. More specifically, ACQUA is installed as a script which is triggered every time a user visits a SE question-webpage. An overview of how ACQUA operates is shown in Figure 6. Upon visitation, A REST request is invoked automatically from the client's browser to the ACQUA server. The request, which contains the URL of the page visited is used to contact the SE API[15] and fetch the content of all answers. ACQUA takes as input the content of all candidate answers and calculates the linguistic features together with their discretised version. The discretised values (see case 3 of previous section) are then passed to a pre-trained classifier. If available, the dedicated SE classifier is used; else the SO classifier is used (as of September 2014, 127 websites are deployed, while ACQUA has 21 classifiers as discussed previously). Ultimately, the classifier returns the answer id which is predicted to be the best answer. Finally, the answer id of the predicted answer is passed to the client's browser where the GreaseMonkey-powered ACQUA script highlights the corresponding text.

Apart from a web application, ACQUA is also provided as a web service. A single method is exposed through an HTTP REST interface on the address http://acqua.kmi.open.ac.uk/predict passing as a *URL* parameter the SE question webpage. In addition to the URL parameter, a cookie with the SE access token (named *sx_access_token*) must also be passed. This is a requirement for ACQUA since the SE API applies a rate limitation of 10,000 requests per day for all third-party applications such as ACQUA. In fact, the same access token is also being used when using ACQUA as a web application: during the installation process, users must grant authentication to the ACQUA application which results in the installation of the above cookie on their browser. Hence, both the web application and the web service invoke the exact same method. Finally, concerning privacy issues, ACQUA is only keeping track of anonymised data and does not use the access token for any other purpose apart from the task of best answer prediction.

7 Discussion

In this section we discuss our results in relation to earlier work, we comment on implications stemming from the proposed methodology and mention potential extensions of this work.

[12] http://stackexchange.com/sites

[13] http://stackoverflow.com/questions/10881047

[14] http://en.wikipedia.org/wiki/Greasemonkey

[15] http://api.stackexchange.com/

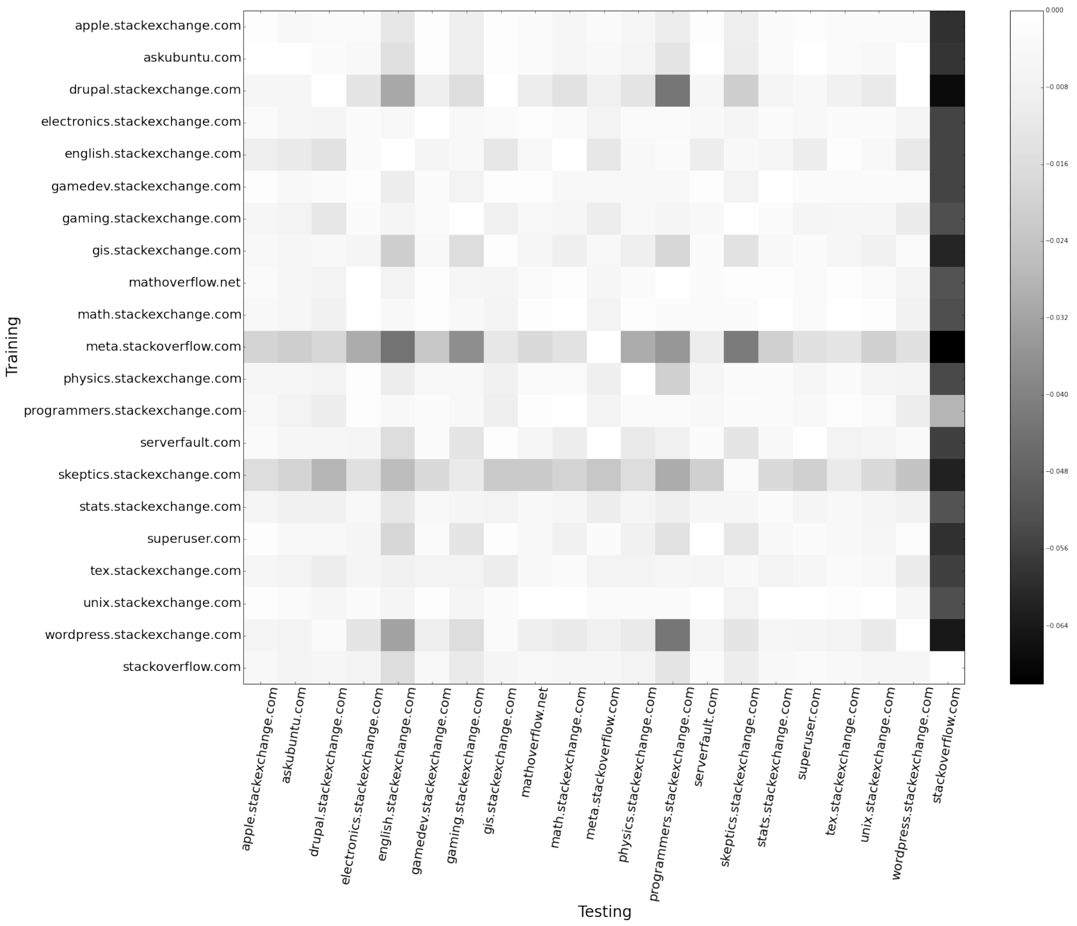
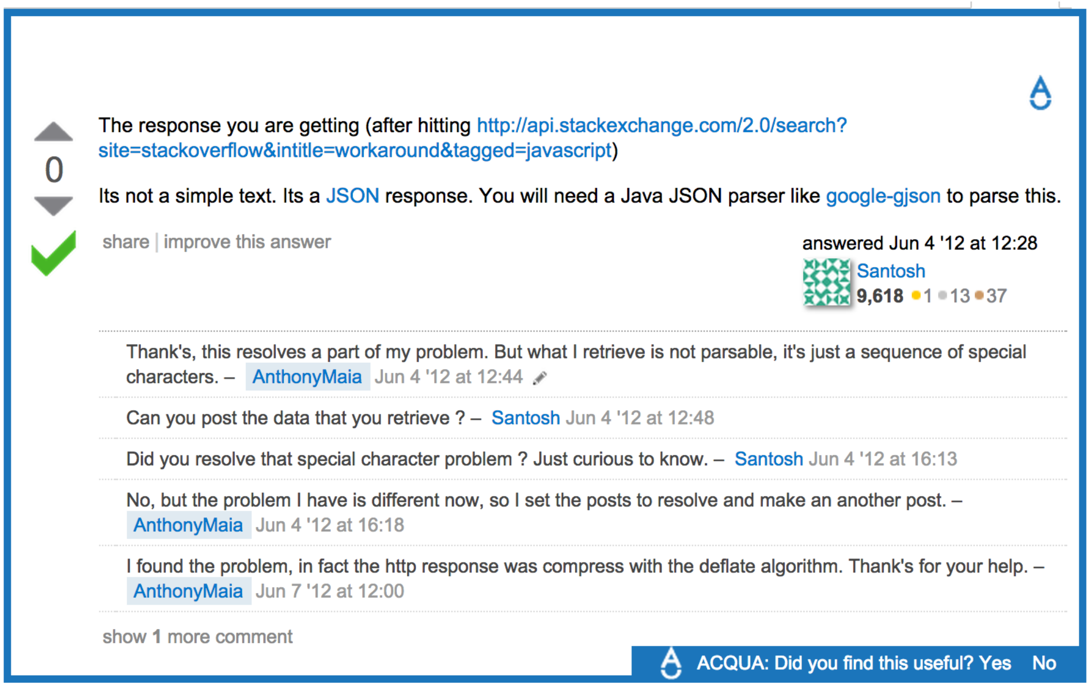


Figure 4: Heatmap with the results for AUC loss when doing paired testing.



The response you are getting (after hitting http://api.stackexchange.com/2.0/search?site=stackoverflow&intitle=workaround&tagged=javascript)

Its not a simple text. Its a JSON response. You will need a Java JSON parser like google-gjson to parse this.

answered Jun 4 '12 at 12:28
Santosh
9,618 ●1 ●13 ●37

Thank's, this resolves a part of my problem. But what I retrieve is not parsable, it's just a sequence of special characters. – AnthonyMaia Jun 4 '12 at 12:44

Can you post the data that you retrieve ? – Santosh Jun 4 '12 at 12:48

Did you resolve that special character problem ? Just curious to know. – Santosh Jun 4 '12 at 16:13

No, but the problem I have is different now, so I set the posts to resolve and make an another post. – AnthonyMaia Jun 4 '12 at 16:18

I found the problem, in fact the http response was compress with the deflate algorithm. Thank's for your help. – AnthonyMaia Jun 7 '12 at 12:00

Figure 5: Screenshot of the ACQUA system. The highlighted answer is from http://stackoverflow.com/questions/10881047.

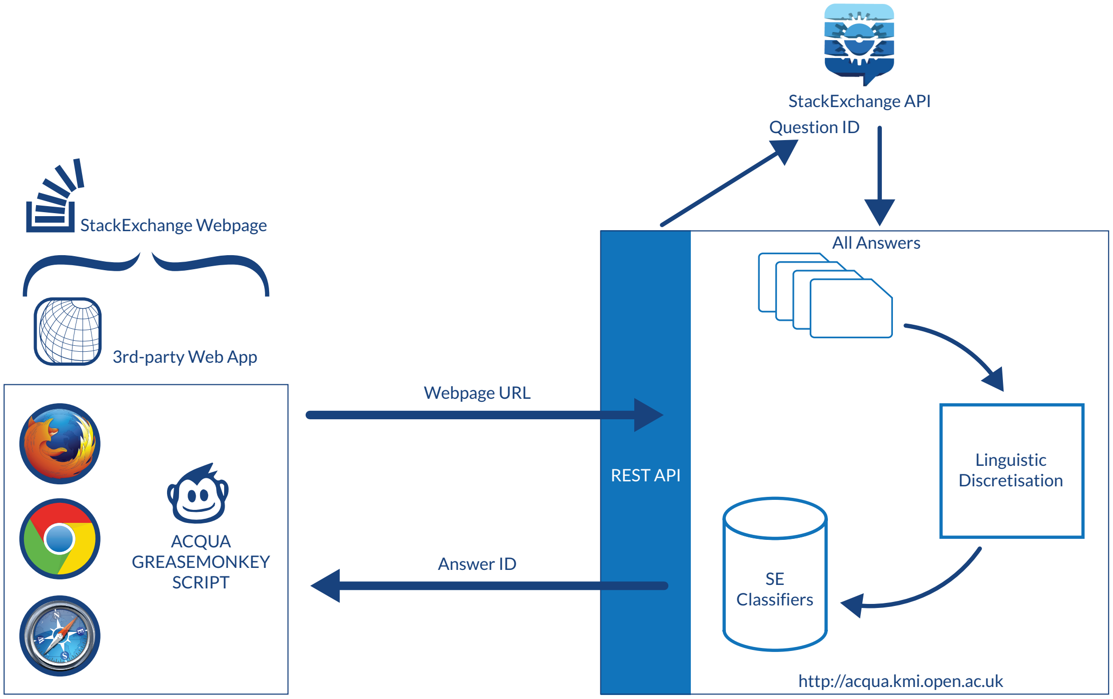


Figure 6: Overview of the ACQUA system.

7.1 *Feature analysis*

Section 5 presented the results of different evaluations aiming at assessing the efficiency of our methodology and understanding how our methodology compares to the state-of-the-art. Hereby, our goal is to analyse the features we proposed and assess their impact on the prediction. To do so, we repeated the experiments using a single feature at a time, starting with the absolute values of linguistic features and continuing with their discretised version. Table 6 presents a summary of macro averages for Precision, Recall, F-Measurement and Area Under the Curve. In this table, every two consecutive rows report the performance of the linguistic feature together with its discretised version. The last row presents the performance when using all features presented in the table. To make the results more readable, we have added four more columns with values showing the relative change from the discretisation process.

Table 6 allows us to make two important observations concerning the selection of our features. The first observation stems by looking at the four columns reporting on the relative change. The discretisation of linguistic features results in the significant increase in performance, since all 5 discretised linguistic features are at least 30% more efficient. More specifically, *Length* is the best feature and – when used alone – manages to deliver only 0.59 AUC (versus 0.78 AUC for $Length_D$, 31.58% AUC increase). $CreationDate_D$, which is not a linguistics-based features and consists the discretised version of *Age*, is the only one producing less than 30% (20% higher AUC). The second observation concerns the performance of prediction when using one feature against using all of them together. The last row in the table shows that using all features results in gaining an important boost over the single-feature performance. This shows that the selection of our features is meaningful and the information contained within them is complementary.

7.2 *Comparison*

As already discussed in Section 2, the paper by Burel *et al.* (2012) predicts accepted answers for Server Fault and Cooking SE. Our work did not include Cooking SE, but we include the larger, more up-to-date dataset of Server Fault (95k vs. 172k answers). Burel et al's classifier based on content delivers a precision of 64.7%, 0.628 F-Measure and 0.679 AUC. Our methodology, which employs discretisation of linguistic features, outperforms their work by 18-21%, since for Server Fault our precision is 83%, F-Measure is 0.74, AUC is 0.85 (Case 3) and 86% precision, 0.69 F-Measure and 0.83 AUC (Case 2). Moreover, their results, when they consider contextual features such as user and answer ratings, are similar to ours achieving the same F-Measure 0.84, with our precision and AUC at 89% (5% higher) and 0.93 (0.02 higher), respectively.

Similarly to us, Tian *et al.* (2013), look at the content of answers. However, they also exploit features related to what we refer to as answer ratings, since they also consider the number of comments to each answer, a feature which is reported as amongst the most informative ones. The authors report a prediction accuracy of 72.27% on a SO dataset of 196k answers at least one year old. By comparison our SO dataset contains 7.1 million answers and our classier returns 82% precision, 0.73 F-Measure and 77% prediction accuracy, which constitutes a noticeable increase in performance.

While the work concerning YA cannot be compared directly to ours, we highlight some analogies and discuss the results. For instance, Shah and Pomerantz (2010) constructed a negative

Table 6: Results for the task of best-answer prediction using exactly *one* feature. Values show macro average.

	Macro Average				Relative Change			
	Precision	Recall	FM	AUC	Precision	Recall	FM	AUC
$CharactersPerWord$	0.53	0.60	0.50	0.53				
$CharactersPerWord_D$	0.66	0.66	0.66	0.70	23.94%	9.82%	32.92%	31.71%
LL	0.49	0.60	0.49	0.53				
LL_D	0.66	0.66	0.66	0.71	36.15%	10.66%	36.76%	32.44%
$Length$	0.58	0.62	0.56	0.59				
$Length_D$	0.75	0.75	0.75	0.78	29.47%	22.12%	33.95%	31.58%
$LongestSentence$	0.54	0.60	0.54	0.57				
$LongestSentence_D$	0.70	0.70	0.70	0.75	30.45%	17.01%	30.62%	32.80%
$WordsPerSentence$	0.52	0.63	0.52	0.54				
$WordsPerSentence_D$	0.68	0.68	0.68	0.73	31.25%	8.44%	30.38%	35.31%
Age	0.53	0.72	0.61	0.61				
$CreationDate_D$	0.71	0.71	0.71	0.74	33.85%	-2.07%	16.71%	20.54%
All features	0.81	0.70	0.74	0.84				

classifier with a dataset comprised of a 1:4 ratio of accepted to non-accepted answers. Adamic *et al.* (2008) consider Programming questions submitted in YA and – similarly to us – disregard the ratings of answers and users. The authors report 72.9% precision, which shows the potential of the linguistic features but that there is also room for improvement. Hence, we can assume that, when using discretised linguistic features, it may be possible to significantly increase performance in the case of YA as well.

7.3 ACQUA user experience

Even though ACQUA has only been deployed for less than a month, we already have a cohort of users, mainly faculty staff from our departments and colleagues who were invited to install and use our service. In the following we discuss some early findings concerning usage, as well as feedback, mostly anecdotal, given to us so far. Our intention is not to evaluate how effective the task of best answer prediction is; we believe this has been addressed in great detail in the previous section. Instead, we aim to report on early feedback that we have received concerning our system.

As of January 2015, ACQUA has been installed 50 times and has served 6042 SE question-webpages. On average, ACQUA serves 34 pages per day. In total, 48 different SE websites have been visited. SO accounts for 85% of the incoming ACQUA requests while 15 SE websites reported just one visit. Concerning the ACQUA user profile, the ones who have installed the ACQUA extension on their browser are already familiar with SE and especially SO. Hence, it was easy for them to generate this amount of requests through “organic” traffic. Concerning the overall experience from ACQUA’s usage, there is a consensus amongst our users that the service is both useful and user-friendly. More specifically, the users are satisfied with the fact that, most of the time, ACQUA is successful in predicting the best answer. Our users assert that they find ACQUA’s response to be in accordance with the actual accepted answer a “surprisingly recurring” incident. In cases where there is a miss-match between ACQUA and the actual SE answer, our users found that either the ACQUA highlighted answer is indeed better, or that it manages to identify an answer of high value. More generally, our users report that the browser experience is enriched, ACQUA is assisting them in browsing the SE websites more efficiently and that they feel more empowered in locating content of high quality.

While the feedback received so far is indeed very promising, we attribute this outcome to various reasons not necessarily linked exclusively to ACQUA itself. First of all, the significance of the above feedback is not solid, since the users of our system are neither a large group nor carefully selected to eliminate possible bias. We expect that more indicative data on usability and usefulness of the system would be collected over a longer period of time. Also, we believe that the power of good quality answers lies largely in the curation of the content that takes place in and by each SE community. In our approach, ACQUA constitutes a mechanism that manages to enhance the overall browsing and information filtering experience in a collection of already high quality answers. The particular factual nature of the discourse in SE (SO describes itself as “all about getting answers. It’s not a discussion forum.”), in combination with the human supervision (e.g. removal of duplicate or inappropriate answers) that takes place, has allowed us to implement the above service in the most meaningful way. However, recent work (Tan *et al.*, 2014) on the effect of wording on message propagation on twitter shows that there is great potential in using approaches to social media analysis motivated by the detection of textual quality.

7.4 Implications from this work

Following the discussion above, it is worth emphasising that ACQUA’s intention is not to eliminate or exclude answers that are not selected as the best. Also, it does not aim to replace the human task of best answer selection. Instead, our position is that ACQUA can be offered as a complementary mechanism for all types of questions, both with or without a best answer. In the case of questions without an accepted/best answer, we consider the primary stakeholders to be the readers-visitors of the website. The authors-repliers also benefit. Since there is a significant proportion of questions without accepted answers, ACQUA “scans” the available answers, proposes the best and allows potential repliers to work on their contribution more efficiently. This is a by-product of making it easier to spot questions that have been answered in a satisfactory manner so that repliers can focus their efforts on other questions. ACQUA can also be quite useful in the case of questions that do have accepted answers chosen by humans; the single-person, one-off selection of the best answer is inherently error-prone and may also become obsolete. As discussed by Oktay *et al.* (2010), a significant amount of answers in SO continue to arrive even after a question has been marked with an accepted answer. The above observation, together with the early feedback we have received from our users, is very encouraging for ACQUA. Even in cases where the decision of the accepted/best answer has already been made once by a human, ACQUA repeats the assessment on the content of all answers (some of which may have arrived after the designation of the human-accepted answer) every time a page is visited and may find more recent answers of potentially higher quality.

8 Conclusions

Previous research on best answer prediction has shown that linguistics-based features can be helpful to a limited extent. The relevant literature shows that features based on user reputation and answer ratings manage to boost the performance of classifiers and outperform purely content-based approaches. Our approach adopts a novel way of processing linguistic features and manages to bridge the above gap. To do so, instead of processing all answers as one solid training dataset, the proposed discretisation process manages to highlight the distinct characteristics of each answer compared to its candidate, "competing" answers. The information that is produced from this process dramatically improves the performance of our classifier. Our extensive evaluation shows that shallow features, such as length and longest sentence, can be very informative, contradicting the findings of earlier work. Hence, encoding this information into a discretised form allows us to train a classifier that is effective enough to match other classifiers that do use and depend upon non-linguistic contextual information.

Our evaluation shows that the performance of our proposed approach matches the performance of reputation-based classification. Contrary to our intuition, the inclusion of more information, such as user background information, does not improve the classification, a sign that reputation information is not independent of information found in linguistic features. Finally, our classification methodology is generic and can be applied to the rest of the SE websites, without the need for training data from the target website. Shallow features, such as answer length and longest sentence, can be used effectively for assessing user-generated text, following our methodology.

To our knowledge, the proposed technique of dealing with continuous and multi-dimensional data found in shallow features constitutes a novel approach for assessing user-generated content. We intend to explore this direction further, explore more linguistic features and features indicative of textual quality, and apply the approach to other social media environments. For example, one direction would be to analyse the linguistic characteristics of different roles in on-line communities, such as initiators, conversationalists, etc. (see for example Angeletou *et al.*, 2011). Another possibility is to follow up on the work conducted by Anderson *et al.* (2012) and explore the assortativity between user reputation and linguistic characteristics of user input. Finally, concerning the ACQUA system, we will explore the possibility of providing an authoring-tool type functionality of predicting the likelihood of an answer being selected as 'accepted/best', in real-time, while it is being authored.

Acknowledgments

This work was supported by the CARRE (No.611140) and COMPOSE (No.317862) projects funded by the European Commission Framework Programme 7.

References

Adamic, L. A., J. Zhang, E. Bakshy, and M. S. Ackerman. 2008. "Knowledge Sharing and Yahoo Answers: Everyone Knows Something". In: *Proceedings of the 17th international conference on World Wide Web*. ACM. 665–674.

Agichtein, E., C. Castillo, D. Donato, A. Gionis, and G. Mishne. 2008. "Finding High-Quality Content in Social Media". In: *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM. 183–194.

Anderson, A., D. Huttenlocher, J. Kleinberg, and J. Leskovec. 2012. "Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 850–858.

Angeletou, S., M. Rowe, and H. Alani. 2011. "Modelling and Analysis of User Behaviour in Online Communities". In: *The Semantic Web–ISWC 2011*. Springer. 35–50.

Burel, G., Y. He, and H. Alani. 2012. "Automatic Identification of Best Answers in Online Enquiry Communities". In: *The Semantic Web: Research and Applications*. Springer. 514–529.

Callan, J. and M. Eskenazi. 2007. "Combining Lexical and Grammatical Features to Improve Readability Measures for First and Second Language Texts". In: *Proceedings of NAACL HLT*. 460–467.

Collins-Thompson, K. 2014. "Computational Assessment of Text Readability: A Survey of Current and Future Research". *International Journal of Applied Linguistics*. 165(2): 97–135.

Danescu, C., R. West, D. Jurafsky, J. Leskovec, and C. Potts. 2013. "No Country for Old Members: User Lifecycle and Linguistic Change in Online Communities". In: *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 307–318.

Feng, L., M. Jansche, M. Huenerfauth, and N. Elhadad. 2010. "A Comparison of Features for Automatic Readability Assessment". In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics. 276–284.

Freund, Y. and L. Mason. 1999. "The alternating decision tree learning algorithm". In: *ICML*. Vol. 99. 124–133.

Gkotsis, G., K. Stepanyan, C. Pedrinaci, J. Domingue, and M. Liakata. 2014. "It's All in the Content: State of the Art Best Answer Prediction Based on Discretisation of Shallow Linguistic Features". In: *Proceedings of the 2014 ACM Conference on Web Science. WebSci '14.* Bloomington, Indiana, USA: ACM. 202–210.

Gunning, R. 1968. "Technique of clear writing".

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. "The WEKA Data Mining Software: An Update". *ACM SIGKDD Explorations Newsletter.* 11(1): 10–18.

Jeon, J., W. B. Croft, J. H. Lee, and S. Park. 2006. "A Framework to Predict the Quality of Answers with Non-Textual Features". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '06.* Seattle, Washington, USA: ACM. 228–235.

Jones, J. and N. Altadonna. 2012. "We Don't Need No Stinkin' Badges: Examining the Social Role of Badges in the Huffington Post". In: *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.* ACM. 249–252.

Liu, J., Q. Wang, C.-Y. Lin, and H.-W. Hon. 2013. "Question Difficulty Estimation in Community Question Answering Services". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.* 85–90.

Liu, Q., E. Agichtein, G. Dror, E. Gabrilovich, Y. Maarek, D. Pelleg, and I. Szpektor. 2011. "Predicting Web Searcher Satisfaction with Existing Community-based Answers". In: *SIGIR.* 415–424.

Louis, A. 2012. "Automatic Metrics for Genre-specific Text Quality". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop.* Association for Computational Linguistics. 54–59.

Louis, A. and A. Nenkova. 2013. "What Makes Writing Great? First Experiments on Article Quality Prediction in the Science Journalism Domain". *Transactions of the Association for Computional Linguistics.* 1: 341–352.

Louis, A. and A. Nenkova. 2014. "Verbose, Laconic or Just Right: A Simple Computational Model of Content Appropriateness under Length Constraints". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics. 636–644.

Oktay, H., B. J. Taylor, and D. D. Jensen. 2010. "Causal Discovery in Social Media Using Quasi-Experimental Designs". In: *Proceedings of the First Workshop on Social Media Analytics.* ACM. 1–9.

Piantadosi, S. T., H. Tily, and E. Gibson. 2011. "Word lengths are optimized for efficient communication". *Proceedings of the National Academy of Sciences.* 108(9): 3526–3529.

Pitler, E. and A. Nenkova. 2008. "Revisiting Readability: A Unified Framework for Predicting Text Quality". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics. 186–195.

Rowe, M., M. Fernandez, S. Angeletou, and H. Alani. 2013. "Community Analysis through Semantic Rules and Role Composition Derivation". *Web Semantics: Science, Services and Agents on the World Wide Web.* 18(1): 31–47.

Shah, C. and J. Pomerantz. 2010. "Evaluating and Predicting Answer Quality in Community QA". In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval.* ACM. 411–418.

Tan, C., L. Lee, and B. Pang. 2014. "The effect of wording on message propagation: Topic- and author-controlled natural experiments on Twitter". In: *Proceedings of ACL.*

Tanaka-Ishii, K., S. Tezuka, and H. Terada. 2010. "Sorting Texts by Readability". *Computational Linguistics.* 36(2): 203–227.

Tian, Q., P. Zhang, and B. Li. 2013. "Towards Predicting the Best Answers in Community-Based Question-Answering Services". In: *Seventh International AAAI Conference on Weblogs and Social Media.*

Vadlapudi, R. and R. Katragadda. 2010. "On Automated Evaluation of Readability of Summaries: Capturing Grammaticality, Focus, Structure and Coherence". In: *Proceedings of the NAACL HLT 2010 Student Research Workshop.* Association for Computational Linguistics. 7–12.

Yang, L., S. Bao, Q. Lin, X. Wu, D. Han, Z. Su, and Y. Yu. 2011. "Analyzing and Predicting Not-Answered Questions in Community-based Question Answering Services." In: *AAAI.*

Yannakoudakis, H. and T. Briscoe. 2012. "Modeling Coherence in ESOL Learner Texts". In: *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP.* Montreal, Canada: Association for Computational Linguistics. 33–43.