On Yahoo Answers, Long Answers are Best

Alina Beygelzimer BEYGEL@YAHOO-INC.COM
Yahoo Labs, New York, NY 10036

Ruggiero Cavallo CAVALLO@YAHOO-INC.COM
Yahoo Labs, New York, NY 10036

Joel Tetreault TETREAUL@YAHOO-INC.COM
Yahoo Labs, New York

Abstract

We provide an analysis of *best answers* (as chosen by questioners) on Yahoo Answers, a popular online Q&A site with millions of monthly contributors. Our analysis is done mainly through the lens of prediction: we compile a dataset that is as large and fine-grained as any considered before, generate features across a range of different classes, and build a classifier to predict which answers will be selected as "best". On the dataset as a whole, despite the breadth and sophistication of our features and learning framework, we achieve virtually no performance edge over the following simple baseline: *choose the longest answer*. Propelled by this unexpected discovery, we perform a detailed analysis of answer length and how it relates to other variables of interest, such as answer time and number of answers. We explore subsets of the data designed to probe into areas where the longest-answer baseline may be handicapped, but we consistently find that it is competitive with our full-featured learner. Our results suggest future directions of study, e.g., controlled experimentation or user interviews, which may shed further light on why answer length is such a good proxy for (the questioner's estimation of) answer quality.

1. Introduction

Community-based Question Answering (CQA) sites such as Yahoo Answers and Naver KiN are popular forums for the casual exchange of information. There have been hundreds of millions of answers posted on Yahoo Answers alone. In several CQA sites, including Yahoo Answers, questioners can select one of the responses as a "best answer". This answer is subsequently ranked first on the page and thus rendered more visible to other users who view the question.

We describe our efforts to build a system to predict—for any given question—which among its submitted answers will be chosen by the questioner on Yahoo Answers.

The most salient, surprising discovery about this task is that simply choosing the *longest* answer to any given question is a good approximation to what can be achieved with a full-bore machine learning approach: A rigorously designed predictor, built using copious training data and leveraging a formidable array of multi-faceted features, can be approximately matched by the simple *longest answer* heuristic. Prior work does not take into account the history of the questioner and answerers (beyond such simple attributes as best-answer rate), and our initial hope was that building broad personalization profiles involving hundreds of features would be useful. It was very surprising that this yielded essentially no benefit.

Our motivation is threefold: first, we would like to understand whether questioner preferences are basically predictable, especially in the context of a system that makes a concerted effort to tailor learning in a personalized way to each questioner or answerer. Second, we hope to understand what factors drive best-answer selection. For instance, if it had turned out that low chronological answer rank is most strongly predictive, that would suggest that either the most qualified answerers are first to respond, or questioners place high priority on very timely responses.

Our result regarding the importance of answer length points in a different direction. It strongly suggests that either questioners have an inherent preference for longer answers, in the aggregate, or that answer length is a remarkably good proxy for whatever complex of features actually drives questioner best-answer selection. We probe the edges of this question through a careful observational analysis of the relationships between some of the most prominent answer features, and indeed a quasi-suggestive picture emerges; but as is almost always the case with pure observation, we are unable to draw any solid conclusions about causality.

The third motivation is more practical: we hope that facility at predicting which answers are most sought by questioners will help improve the CQA platforms themselves.[1]

There is an abundance of prior work considering various dimensions of the best-answer prediction problem (see Appendix A). Unfortunately, there is no agreed upon set of baselines used in the literature, making cross-paper comparisons difficult, especially when different data sets are used. And while many papers include the answer length as one of the features, there is typically no explicit comparison to simply choosing the longest answer. Our recommendation to the research community is to become more consistent in its use of baselines, as well as in the reporting of data filtering and evaluation principles. The answer length baseline should be a prominent standard of comparison in all closely related future studies.

2. Dataset

Our data is drawn exclusively from contributions to Yahoo Answers during 2012 and early 2013. We start with a set of questions and *all* answers they received, and perform the following filtering:

- We filter out all questions (and their answers) that received at least one non-English answer.
- We filter out any answers that were marked as deleted.
- We filter out any questions that do not have a best answer selection.
- We filter out any answers that were submitted *after* the questioner made his best-answer selection, since they could not possibly have been selected.

Modulo these filters, our dataset is a representative sample of Yahoo Answers content, given the time period from which it is drawn from. For each question/answer pair we extract a rich set of attributes, including the text, timestamps, the number of "points" held by the users at the time the answer was submitted, and more; we also have identifiers for the questioner and answerer that allow us to compute aggregates over the training data to be deployed as features at test time.

Our final dataset consists of approximately 5.6 million questions and 26 million answers. We split the data into a training set of size 5.1 million questions, and a test set of size 461K questions. However, for prediction we found that there is actually no performance difference (< 0.001 percent accuracy loss) if we use only a subset of approximately 10% of this data (further split 90%/10% for training/test). Our reports in the Results section are based on that smaller subset.

3. Statistical analysis

We begin our study with an expository analysis of the Yahoo Answers dataset as a whole, with a focus on factors germane to the picture revealed by our predictive analysis. We will provide an overview of the distribution over number of answers per question, answer times, and answer lengths; and, in light of the dominance of answer length we ultimately see in prediction, we attempt to partially disentangle length from other variables as a driver of best-answer selection.

3.1. Number and timing of answers and best-answers

Because of the diversity of question types that appear on Answers, and the variance in question popularity, there is a broad distribution over the number of answers received. For instance, it is not very uncommon for poll-style questions to receive over one hundred answers, whereas many factoid-style questions receive only one or two. The most-answered question[2] in our training set received over 300 answers, polling readers about the last time they went to a particular restaurant. The distribution through 18 answers-per-question is given in Figure 1. The average number of answers across the training set is 4.66, and the median is 3.

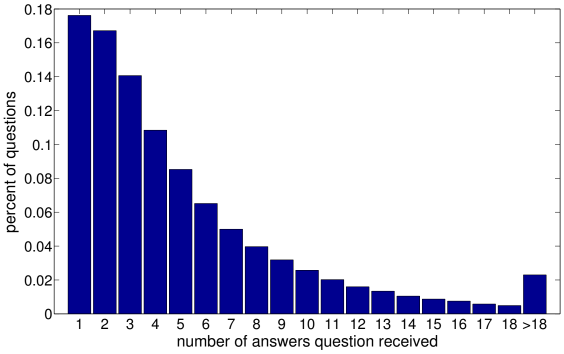

Figure 1. Histogram of number of answers received, across all questions in the training set.

When we look at the distribution of *best* answers across chronological ranks (i.e., first-answer, second-answer, etc.) we see a strong increasing pattern, regardless of the number of answers: later "positions" have progressively higher best-answer rates. This is illustrated for answer-sizes 2, 5,

[1] For instance, if we could predict—for any given questioner and hypothetical answerer—the odds that the answerer would give the "best answer" to the question, that would allow us to more efficiently shepherd answerers to the questions they match best. This functionality could be realized in search tools and interactive answerer guidance tools.

[2] That is, other than those asked by Yahoo Answers official accounts, which are highly promoted and may receive thousands of answers.

10, and 15 in Figure 2.

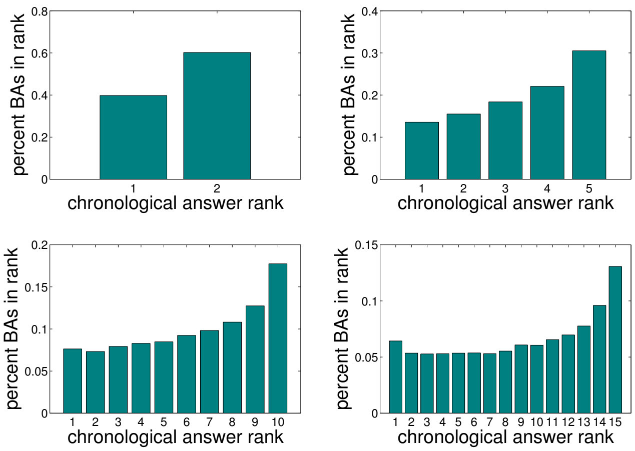

Figure 2. Chronological position of best-answers across the training set, sliced by number-of-answers subsets. E.g., the top-left picture illustrates the fact that, among all questions that received two answers, the later answer was chosen as best about 60% of the time.

3.1.1. Answer timing

In general, questions are answered quickly. The median elapsed time between question and first-answer is on the scale of minutes (though the average is one or two orders of magnitude larger, as a minority of eventually-answered questions remain unanswered for very long periods). Moreover, most questions receive their final answer (that is, final prior to best-answer selection) within hours. Best-answer selection, which is always at least an hour after question-time, typically occurs close to a day later. Figure 3 illustrates the entire distribution over first-answer, last-answer, and best-answer-selection times.

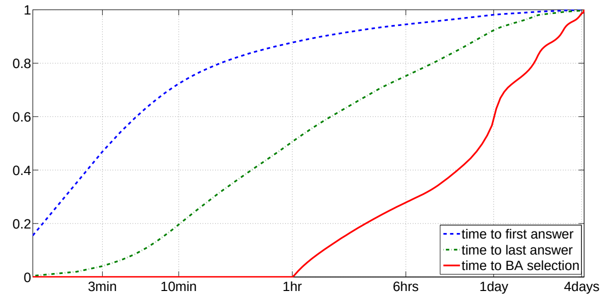

Figure 3. Cumulative distributions for elapsed time between question and: first answer, last answer, and best-answer-selection in the test set. Note that the x-axis is logscale.

3.1.2. Last answer bias

There are several hypotheses one could formulate to try to explain the stark preference for last-answers illustrated in Figure 2. Perhaps better-quality answers appear at the end, because answerers contribute only to those questions that appear to "need" a better answer (either seeking points via best-answer award, or simply seeking to help). On the other hand, since answers were presented in chronological order on the Answers web page for all of the data in this study, there could be some inherent questioner bias towards selecting answers further down on the page, especially the last-appearing one.

The way to definitively discriminate amongst these theories would be a randomized experiment in which presentation orders are manipulated. We were unable to do that, but we try to at least get near the issue in another way: we look at the respective best-answer odds of consecutive answers to the same question, as a function of the elapsed time between the two answers. If there is an inherent bias towards answers with larger chronological rank (due to the presentation, say), this should be manifest even if the time interval between answers is nearly 0. But we do not observe that in the data; in fact there is a slight negative bias near 0.

Figure 4 illustrates that the preference for the later answer rises starkly as a function of the time gap between answers. We consider every consecutive pair of answers (A_1, A_2) to every question in the test-set, with A_2 submitted after A_1; we then bin pairs by decile $\beta_1, \ldots, \beta_{10}$ of time$(A_2)-$time(A_1), and then for each bin $\beta \in \{\beta_1, \ldots, \beta_{10}\}$ we compute:

$$y(\beta) = 100 \cdot \frac{\sum_{(A_1,A_2)\in\beta} \mathrm{BA}(A_2) - \sum_{(A_1,A_2)\in\beta} \mathrm{BA}(A_1)}{\sum_{(A_1,A_2)\in\beta} \mathrm{BA}(A_1)},$$

where $\mathrm{BA}(A)$ is 1 if A was selected as best-answer and 0 otherwise.

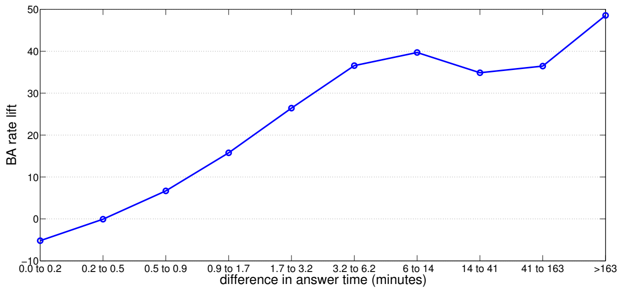

Figure 4. Advantage (in terms of best-answer likelihood) as a function of elapsed time since the preceding answer. Computed on the test-set.

3.2. Answer length

Just as the number and timing of answers varies widely, sc does the length of answers. Across our training set the average answer length is 43 words (the median is 20).[3] Bu there are classes of questions—such as polls—that receive very many, very short answers; other classes—perhaps personal advice-style answers—may receive fewer but longe: answers.

By far the most salient fact that emerges from studying the best-answers data is that *longer answers are more ofter best-answers* (hence this paper's title). Figure 5 illustrate: the respective answer-length distributions for the entire se of answers, and also for just the best-answers.

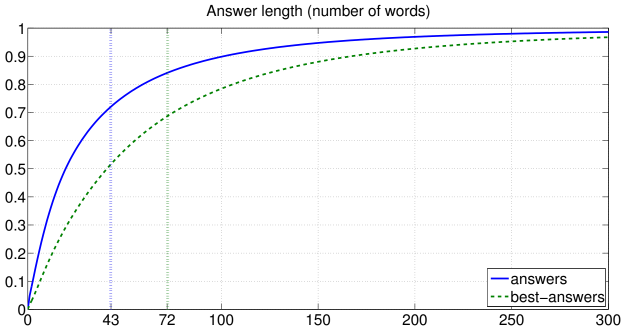

Figure 5. Cumulative distribution of answer and best-answer lengths (in words) across the training set. Average answer length across all answers is 43 words; it is 72 for best answers. Answers at least 100 words long make up more than twice the percentage of *best* answers as they do *all* answers (21% vs. 10%). A third of answers—but only a sixth of *best* answers—are less than 10 words long.

Figure 6 provides another view, sliced by number-of-answers the question received. It is an analog of Figure 2, where instead of ranking by chronology (most recent first), we rank by answer-length (shortest first).

3.2.1. Later (first) answers are longer

In light of the salience of answer length as a factor that distinguishes best from non-best answers, we will now take some time to drill into the relationship of length to other factors, particularly time (chronological rank and elapsed time to answer).

We first observe that answer length grows dramatically with elapsed time to first-answer. More specifically, it rises sharply for the first several deciles, and then plateaus (see Figure 7). One possible explanation for this is an answerer selection bias: new questions are given prominence on the Answers site[4] and hence at first may attract a relatively wide-ranging audience; however, after the initial recency-based prominence of the question dies away, it is likely to be viewed only by those who have happened upon it through directed search (or perhaps via "related questions" links from other questions). Those later answerers are likely to be more interested in answering this question *specifically*, and thus may take more time to write more detailed answers.

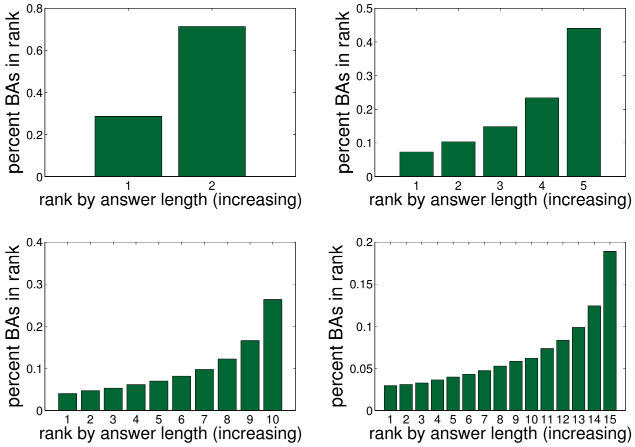

Figure 6. Answer-length rank (i.e., by number of words, least to greatest) of best-answers across the training set, sliced by number-of-answers subsets. E.g., the top-left picture illustrates the fact that, among all questions that received two answers, the longer one was chosen as *best* about 70% of the time.

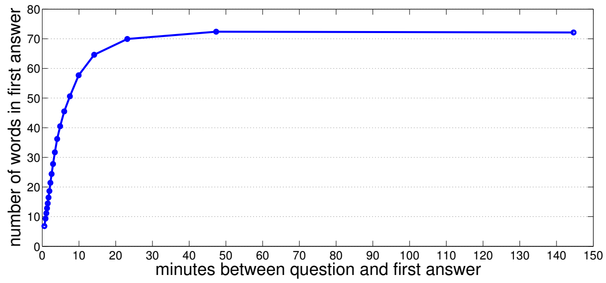

Figure 7. Relationship between time-to-first-answer and answer-length. Each data point is the average of 244K answers across the training set, with the 5 percent of all answers with longest time-to-first-answer excluded.

[3] For our purposes, the definition of a "word" is any text separated by whitespace; so things like emoticons and floating punctuation will be included in the word count.

[4] During the time period from which the data was collected, users could choose to browse "recent" questions on the main page.

3.2.2. Longer (first) answers have fewer follow-up answers

Strongly connected to the above (and Figure 7) is Figure 8, illustrating how the likelihood of being the best (and the *only*) answer changes as a function of the *first* answer's length and time. In 90% of cases, short (first) answers of less than 10 or so words have at least one follow-up answer. For questions whose first answer is long (over 200 words), the percentage drops to about 50.

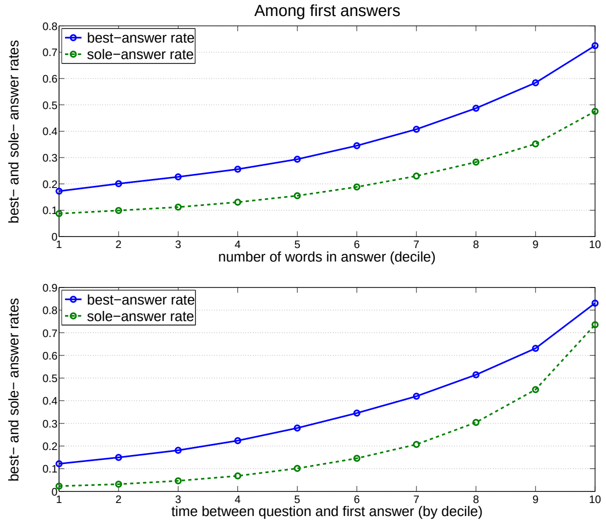

Figure 8. Best- and sole-answer rates as a function of answer length and answer time, among *first* answers. Each data point is the average of 10% of the answers (513K) across the entire training set.

Figure 7 tells us that after an initial period, as the time-to-first-answer increases, the first-answer length does not increase—on average. But late-arriving first answers will of course still have a large variance in answer length; do longer answers perform better in that subset? Yes. Figure 9 illustrates that, although the best-answer rate is high for all late-arriving first answers—lateness alone is enough to make it improbable that a competing answer will ever come in—the best-answer rate continues to climb with answer length (from about 0.75 for the bottom decile to 0.95 for the top).

The likelihood of being the only answer follows a similar trend. Unfortunately we cannot say whether this is due to longer answers "scaring off" further answers, or simply a selection bias regarding the type of questions that receive long answers (e.g, it may be that long answers are natural for questions that are less likely to attract interest).

3.2.3. "Controlling" for answer time

We see two very clear trends: an increase in best-answer rate as answer time and answer length increase. Do both of these factors individually contribute to answer "quality", or is one just an incidental correlate of the other? That is, are late answerers often best only inasmuch as they tend to be longer? Or are long answers often best only inasmuch as they tend to be later? Or neither? Again, we will be unable to get any actual *causal* conclusions in the absence of controlled experimentation (a third variable may be causally driving the apparent "effect" of both of these variables); but isolating variables can certainly point towards directions for improved prediction.

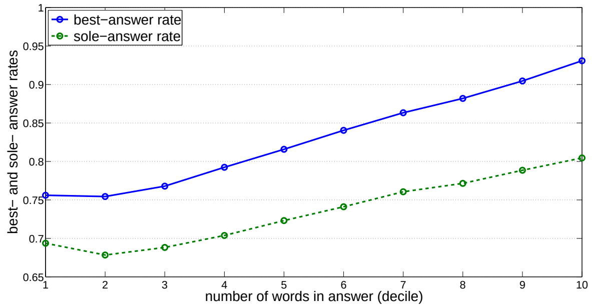

Figure 9. Best- and sole-answer rates as a function of answer length and answer time, among the 10% of first answers that arrived *latest* with respect to their respective questions (more than about an hour after question-time), across the entire training set. Each data point is the average of 10% of the answers (51K) within this subset.

To try to isolate answer *length* from answer *time*, we do a within-question analysis. Specifically, we look at consecutive pairs of answers *to the same question* that were submitted at virtually the same time (within 20 seconds of each other), and then measure the "BA rate lift" as a function of answer length. That is, analogously to the exercise for Figure 4, we consider every pair of answers (A_1, A_2) meeting the timing criterion, letting A_2 be the one of greater length, and we bin pairs by decile β of length$(A_2)-$length(A_1); then for each bin β we compute:

$$y(\beta) = 100 \cdot \frac{\sum_{(A_1,A_2)\in\beta} \mathrm{BA}(A_2) - \sum_{(A_1,A_2)\in\beta} \mathrm{BA}(A_1)}{\sum_{(A_1,A_2)\in\beta} \mathrm{BA}(A_1)},$$

where $\mathrm{BA}(A)$ is 1 if A was selected as best-answer and 0 otherwise. This is BA-rate lift, plotted for each length-difference decile in Figure 10. *A major correlation between length and best-answer selection remains.*

3.2.4. "Controlling" for answer length

We do a similar analysis where answer length is held constant and best-answer rate is measured as a function of answer time (more specifically, *chronological rank*). The analysis is identical to the above, except pairs A_1 and A_2 to the same question are now chosen if their length differs

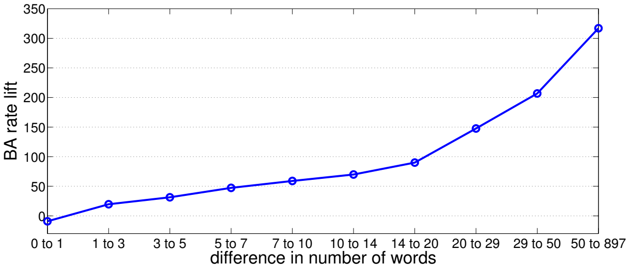

Figure 10. Across the entire training set, measured over all pairs of answers submitted to the same question within 20 seconds of each other (25K per decile datapoint). Length of the first and second answers (chronologically) in each pair differed by less than one word on average (23.8 vs. 24.5), suggesting that the 20 second time threshold is short enough to eliminate any significant chronological bias.

by at most $x\%$ (where we vary x). We measure the BA-rate lift, as described in the preceding formula, as a function of the difference in chronological rank between A_2 and A_1, letting A_2 denote the later answer.

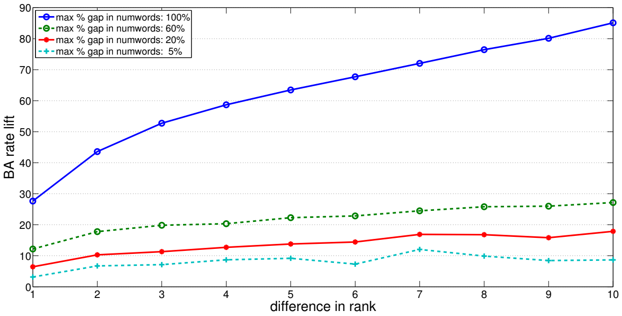

Figure 11. Across the entire training set, measured over all pairs of answers submitted to the same question, meeting various criteria of answer-length "closeness" (a minimum of 4.7K pairs were averaged per datapoint). The x-axis measures how many chronological positions later one of the answers is than the other.

The results are in Figure 11. The top line is the "uncontrolled" plot where we average over all pairs. The other lines in the plot denote the BA-lift for various "tightness settings" defining the criterion for pairs of answers (to the same question) to be included in the averages. The bottommost line includes only answer pairs whose lengths differ by at most 5% (so even a pair with respective lengths 100 and 94 would be excluded). We see that the BA-rate lift largely disappears as answer-length is taken out of the equation. But again, since we're aggregating across questions, we cannot say whether what we observe is primarily a question-selection effect driven by the various answer-length constraints; however, that would be a rather unintuitive distinction in the data, i.e., that some question classes elicit answers of comparable length while others do not.

4. Prediction methodology

Table 1 gives a detailed list of features used to build a predictor. The features are aggregated into three main classes: functional, linguistic and personalization, summarized below.

Functional features include aspects about the questioner and/or answerer such as the number of points at time of asking or answering. For answerers specifically, we also include their chronological rank and the elapsed time since the question and since the first answer.

Linguistic features were motivated by prior work in CQA, such as (Surdeanu et al., 2011), as well as work dealing with text quality, such as automatic essay scoring (Attali & Burstein, 2006) and noisy data processing (Mohammad et al., 2013). "Low level" features include n-grams (both weighted and unweighted), and counts of punctuation, spelling errors and stop words, among others. We also included features which count the number of discourse connectives (Pitler & Nenkova, 2009) (inspired by recent work in measuring discourse complexity (Jansen et al., 2014)), as well as many others (see appendix)

In an effort to leverage the breadth of data we have available, spanning an entire year, we build a set of **personalization features** that aggregate statistics for specific users across the entire training set. For *questioners*, we track the number of questions asked and average number of answers received, and then compute average differences between answers they selected as *best* and answers they didn't, across 23 different dimensions (mostly linguistic and similarity features from above). For *answerers* we do something similar. We compute the total number of questions answered, best-answer awards received, as well as normalized *quality metrics* such as best-answer ratio and various normalizations of such. We compute aggregates of question features (such as question length, questioner points, etc.), separated between the cases where the answerer was selected as best and those where he was not, and we add features that compute the differences over these two sets. Finally, we compute a set of superlative features to identify "top answerers".

Intuitively, having good "personalized" data for answerers seems more important than for questioners, since our task essentially boils down to discriminating amongst answerers on a question-by-question basis. But do the same answerers contribute to the site for a long enough period of time to allow us to accumulate good personalized data to go on? Figure 12 demonstrates that the answer is yes. About 50% of test-set answers came from users who had answered at least 100 questions in the training set. 10% came from

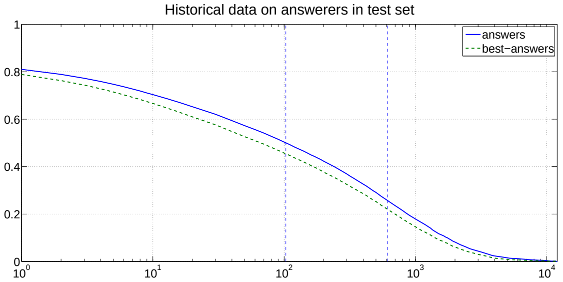

Figure 12. Percentage of answers (and best-answers) in the test set contributed by answerers who answered at least x questions in the training set, from which personalization features were harvested. Note that the x-axis is logscale. Across all answers in the test set, the average number of answers in the training data for the provider of the test-set answer is 611; the median is 103.

those who had answer at least 1750.

4.1. Learning paradigm

We used two open source machine learning systems, Vowpal Wabbit (VW)[5] and scikit-learn.[6]

VW is based on sparse stochastic gradient descent (Duchi et al., 2011; McMahan & Streeter, 2010; Karampatziakis & Langford, 2011; Ross et al., 2013), and is state of the art in large-scale learning. While based primarily on linear learning, it also supports several modes of non-linear learning. It is also a particularly useful data exploration framework, as it natively supports manipulations with groups of features (called namespaces), allowing us to inspect individual namespaces and to include namespace interactions into the feature space. Within scikit-learn, we used gradient boosted regression trees (GBRT) and random forests.

We experimented with the several ways of formulating the learning problem:

1. Encode each question and all its answers as a single example for the learner, with the learner predicting the chronological rank of the best chosen answer. This is a multiclass classification problem.
2. Encode each question and answer pair as a single example, predicting whether or not the answer is chosen as best answer for the question. To make a prediction, we apply the model to each candidate answer, rank them by the predicted score, and choose the answer with the highest score.
3. Train the learner on *pairs* of answers to the same question, with one answer in the pair being the chosen answer, to predict which of the two answers was chosen. To make a prediction, we apply the model to all pairs of answers and choose the answer with the largest number of pairwise wins, breaking ties randomly.

In the first formulation, features associated with answers of different ranks don't share their weights in the model. For example, the feature representing the number of words in the (chronologically) first answer is associated with a different weight than the feature representing the number of words in the second. In the second formulation, there is a single set of weights shared among all answers regardless of their rank. Separating the weights based on the rank can potentially lead to a richer model. For example, it could let the model express that the length of an answer is very predictive but only for the first ranked answer. In our experiments, it turned out that this additional representational power did not yield any statistical improvement.

In our experiments, all of these formulations resulted in roughly the same accuracy. The third (ranking) formulation slightly outperformed the rest, but the gap was only 0.25% (a quarter of 1%) relative difference. We found VW defaults to be reliable on this dataset; any lift we got from tuning parameters was minimal.

5. Results

We now describe the accuracy of our predictor on slices of the data determined by the number of answers each question received. We trained and optimized for each slice, and compare performance against the following baselines:

- *Random*: choose each answer with equal probability
- *Last answer*: always choose last answer
- *Most prior points*: choose answer submitted by user with most points at answer time
- *Historical best-answer ratio*: choose answer submitted by user with largest fraction of past answers chosen "best"
- *Longest answer*: choose answer with most words

Last answer is included because, for every number of answers k, the most common chronological "rank" of the best-answer is k by a significant margin (as evidenced by Figure 2). Figure 6 similarly motivates the *longest answer* baseline.

For each slice, we built our predictor greedily. We started by finding the best single namespace, and then greedily extended the predictor until adding new features was no longer improving the performance on the hold out set. Although the precise set of namespaces that was optimal for

[5] http://hunch.net/~vw/
[6] http://scikit-learn.org/

each number-of-answer slice varied, we know that the number of words feature alone would be approximately optimal in all cases, given the performance of the *answer length* baseline. In addition to answer length, other features which were predictive included n-grams, number of points the Answerer had, and the namespace which consisted of number of stop words, discourse connectives, etc.

Figure 13 illustrates our results. For every slice, the performance hierarchy amongst baselines and our predictor is the same: *random* is the worst, then *most prior points*, then there is a significant gap moving to *historical best-answer ratio* and *last answer*; finally, there is a large gap between the other baselines and *longest answer*, which achieves performance just under that of our learner.

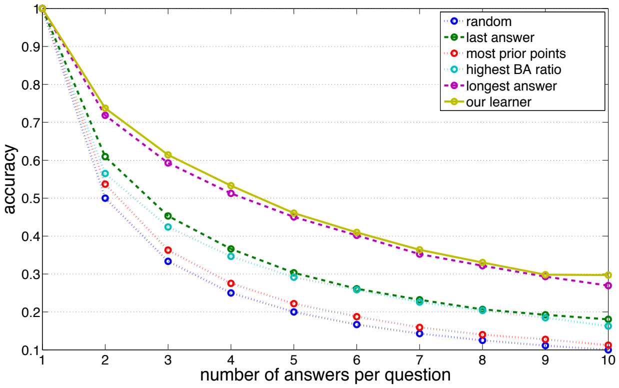

Figure 13. The accuracy of our learner on the test-set, compared to an array of baselines. A predictor based only on answer-length is remarkably competitive with the system performance.

The aggregate performance across all questions in the test set receiving no more than 10 answers (combining all number-of-answers slices ≤ 10) is as follows:

predictor	accuracy rate
random	0.4667
most prior points	0.4872
last answer	0.5477
highest BA ratio	0.5294
longest answer	0.6456
our learner	**0.6583**

The upshot is that simply using the length of the answer is a surprisingly powerful feature and baseline. Questions with no more than 10 answers make up more than 90% of the entire dataset.

The results obtained using scikit-learn were very similar. (We used built-in grid search to optimize parameters as well.) The best accuracy result on for gradient boosting was 0.650, with random forests coming close at 0.639. Per-slice AUC values were also very similar among the different learning algorithms, with AUC being 0.73 for questions with at most 5 answers, and gradually dropping to 0.70 for questions with 10 answers.

5.1. Crippling the longest-answer predictor

To examine how robust the dominance of the longest-answer predictor is, we constructed sub-datasets intended to make discrimination based on answer-length difficult. We went about this in two different ways, described below. In each case, though the performance of the longest-answer predictor was severely diminished, it remained remarkably competitive with our full-featured learner. Details are in Appendix C.

5.2. Evaluation on factual questions

We also constructed a data subset consisting only of questions in the Yahoo Answers category *Math & Sciences*. Our purpose here is to examine a segment of the data that is primarily objective and technical. As expected, best-answers on this segment are more predictable than in the broader dataset. This also corroborates the finding in (Aslay et al., 2013) where performance in the expert-finding task was substantially higher for their system and baselines on factual questions. However, a major part of this performance boost can be accounted for simply by the lower average number of answers per question (2.43 vs. 4.66 for the entire training set). The performance difference between our learner and the longest answer baseline is still negligible.

predictor	accuracy rate
random	0.6417
last answer	0.6907
most prior points	0.6746
highest BA ratio	0.7102
longest answer	0.7743
our learner	**0.7793**

5.3. Effect of training data size

Finally, we wanted to investigate the impact of varying the size of the training set. We compared the performance of our predictor when trained on 10% of the complete training set with that when trained on the entire training set. The performance differed by less than 0.01%, i.e., less than 500 out of 500K questions were correctly classified via the large training set but not via the small one. One of our prior expectations in this work was that access to an extraordinarily large dataset would yield a comparative advantage in learning; that hypothesis is not supported by this result.

References

Agichtein, Eugene, Castillo, Carlos, Donato, Debora, Gionis, Aristides, and Mishne, Gilad. Finding high-quality content in social media. In *Web Search and Data Mining (WSDM)*, 2008. URL http://www.mathcs.emory.edu/~eugene/papers/wsdm2008quality.pdf.

Aslay, Cigdem, O'Hare, Neil, Aiello, Luca Maria, and James, Alejandro. Competition-based networks for expert finding. In *SIGIR*, 2013.

Attali, Yigal and Burstein, Jill. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3), 2006.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Feng, Vanessa Wei and Hirst, Graeme. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 60–68, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P12-1007.

Jansen, Peter, Surdeanu, Mihai, and Clark, Peter. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 977–986, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-1092.

Jurczyk, P. and Agichtein, E. Discovering authorities in question answer communities by using link analysis. In *CIKM*, pp. 919–922. ACM, 2007.

Karampatziakis, Nikos and Langford, John. Online importance weight aware updates. In *UAI*, pp. 392–399, 2011.

Liu, Yandong and Agichtein, Eugene. You've got answers: Towards personalized models for predicting success in community question answering. In *Proceedings of ACL-08: HLT, Short Papers*, pp. 97–100, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-2025.

Liu, Yandong, Bian, J., and Agichtein, Eugene. Predicting information seeker satisfaction in community question answering. In *Proceedings of SIGIR*, 2008.

McMahan, H. Brendan and Streeter, Matthew J. Adaptive bound optimization for online convex optimization. In *COLT*, pp. 244–256, 2010.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

Mohammad, Saif M., Kiritchenko, Svetlana, and Zhu, Xiaodan. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.

Pitler, Emily and Nenkova, Ani. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 13–16, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P09/P09-2004.

Ross, Stéphane, Mineiro, Paul, and Langford, John. Normalized online learning. In *UAI*, 2013.

Shah, Chirag and Pomerantz, Jefferey. Evaluating and predicting answer quality in community qa. In *In Proc. of the 33rd Intl. Conf. on Research and development in information retrieval, SIGIR 10*, pp. 411–418. ACM, 2010.

Surdeanu, Mihai, Ciaramita, Massimiliano, and Zaragoza, Hugo. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2), 2011. URL http://www.aclweb.org/anthology/J/J11/J11-2003.pdf.

Yih, Wen-tau, Chang, Ming-Wei, Meek, Christopher, and Pastusiak, Andrzej. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1744–1753, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-1171.

A. Background

CQA research has been a thriving area in recent years. Prior work has not only looked into best answer prediction but related tasks such as predicting asker satisfaction or automatic reranking of answers. Research into these areas not only gives the scientific community insight into user behavior, but it also paves the way for complex NLP-based automated question-answer systems.

Unfortunately, there is no agreed upon set of baselines used in the literature, making cross-paper comparisons difficult, especially when different data sets are used. It is also often unclear what filtering was used on the data. For example, in best answer prediction, one will of course obtain a higher accuracy score if questions with a single answer are included rather than filtered out. These issues with reporting make it difficult even to assess baselines. Thus, part of the motivation for this work is to drill down and provide an analysis of features possibly taken for granted in some prior works. We now detail some of the closely related prior work.

Early work One of the first works in BAP is (Shah & Pomerantz, 2010). The authors used linear regression with simple linguistic and profile features to predict the best answer in a curated set of questions, all with five answers. (Agichtein et al., 2008) used a host of linguistic features in addition to information about user statistics and networks to classify answers from Yahoo Answers as high or low quality. While the paper does not tackle the task of best answer prediction, its is one of the few to highlight the importance of answer length, which was the dominant feature for their task. The authors found that adding other linguistic and user-oriented features further improved performance.

NLP-focused work Probably one of the most comprehensive NLP works in CQA is (Surdeanu et al., 2011) which utilized a massive array of features including lexical, semantic role labeling, parsing, and n-gram features, as well as different similarity metrics, to rerank a set of answers to a question. They found that the more complex features derived from semantic role labeling and dependency parses were effective for the task of reranking in non-factoid questions.

Ref.(Jansen et al., 2014) noted that prior work into answer reranking relied mostly on intra-sentential or word features, but that answers frequently have many interacting sentences. They showed that features derived from a discourse parser (Feng & Hirst, 2012), in addition to lexical semantic features which model the meaning of the answer, outperform shallower NLP features on the task of reranking answers to non-factoid "how" questions.

In both of the aforementioned works, the focus was on answer reranking rather than BAP. The common baseline to use in this case is the BM25 ranking function from the IR community. In (Jansen et al., 2014) this baseline performed at 41.12% accuracy for BAP (P@1) and the overall system performed at 50.91%, however no comparisons were done to single feature baseline predictors such as answer-length or chronological-rank (simply picking the last answer).

Ref.(Yih et al., 2013) showed that lexical semantics was useful in answer sentence selection, where a system is given a question and a set of candidate sentences, and must choose the sentence with the correct answer. Their baseline consisted of the number of words shared between the question and answer. In our work, we also adopt a lexical semantics approach based on word vector representations to measure the similarity between the question and answer.

Network-motivated methods Another popular research thrust is the discovery and use of the network representing relationships between users in CQA forums (Jurczyk & Agichtein, 2007). Recent work (Aslay et al., 2013) showed that building a network of answerers modeling competition between the best-answerer and other answerers outperformed other methods on the task of identifying expert answerers. However, it did not outperform baselines of best answer ratio and best answer count.

Incorporating personalization features Some prior work (Liu & Agichtein, 2008; Liu et al., 2008) explores the use of personalization features for the task of predicting an asker's "satisfaction"—an asker is considered satisfied if he or she has closed the question and rated the best answer with at least three stars. These works treat the task as a supervised classification problem and use features such as: the wh-type and category of the question; linguistic overlap between the question and each answer, n-grams for the question and answers, functional features such as length of an answer and number of answers, as well as the number of thumbs up or down given by other community members. Personalization features, such as those which model prior overall satisfaction, satisfaction per category, number of questions posted, etc., were found to be useful in boosting performance above baselines. A marked improvement was found for questioners with over 20 questions. Our work differs in that, first, it is on a different task (BAP), and second, we employ a much wider array of personalization features across both askers and answerers.

B. Feature Description

Table 1 below gives a detailed description of features used for prediction.

Question Type is derived using regular expression patterns similar to those used in (Surdeanu et al., 2011) where a question is labeled as one of: *who, what, where, why, when, which, how, be, have, modal*, or *none-of-the-above*. Our hypothesis is that different types of questions will elicit different types or forms of answers. **Question Category** was derived from the Yahoo internal taxonomy of over a 1000 question categories. Our hypothesis is that different topics may attract different forms of answers, or different answerers.

Similarity features are employed to measure the meaning or content of the answer with respect to the question. First, we employ cosine and Jaccard similarity measures comparing the bag-of-words representations of the question and answer. We also compute a distributional similarity measure using the word2vec package (Mikolov et al., 2013),[7] which allows one to compute a vector representations for answer and question strings and then compute a cosine similarity measure between the two vectors. In short, we designed features which are aimed at capturing aspects of text quality, complexity and content.

C. Crippling the longest-answer predictor

To examine how robust the dominance of the longest-answer predictor is, we constructed sub-datasets intended to make discrimination based on answer-length difficult. We went about this in two different ways, described below. In each case, though the performance of the longest-answer predictor was severely diminished, it remained remarkably competitive with our full-featured learner.

Details are in the appendix, but surprisingly, the performance difference is still quite small.

C.1. Two long answers

We pruned the training and test sets to exclude any question in which the lengths of the longest two answers differed by more than 20%. For instance, a question with answers of length 100, 82, 50, and 48 would appear in this subset, but one with answers of length 100, 79, and 60 would not.

C.2. Three long answers

We also considered a variant in which we pruned the data so as to contain only questions with *three* answers of comparable length, although we loosened the allowable gap from 20% to 30% in the interests of getting a large enough dataset. In this subset, a question with answers of length 100, 82, 72, and 4 would appear, but one with answers of length 100, 99, and 60 would not. Our results:

predictor	**accuracy rate: two long answers**	**accuracy rate: three long answers**
random	0.2484	0.1823
last answer	0.3295	0.2581
most prior points	0.2675	0.1933
highest BA ratio	0.3284	0.2508
longest answer	0.4043	0.2941
our learner	**0.4277**	**0.3161**

In both cases the performance drops relative to the full dataset, while our learner's relative edge over the baselines increases slightly. Surprisingly, the performance difference is still quite small—about 2% in each condition.

[7] https://code.google.com/p/word2vec/

Functional features
user ID # number of points at question time (log and τ) A: rank of answer (order), elapsed time between question and answer (log), and elapsed time since first answer (log) Q: question category, question type, # of answers received
Linguistic features
n-grams of text A: weighted n-grams average word length, # of words (τ), one letter words (n), and capitalizations, # of words with non-alpha characters in middle, instances of 3+ repeated characters, and URLs in string, # of punctuations (n), periods (n), question marks (n), and quotation marks (n) # of stopwords, discourse connectives, modals, polite words prepositions, and spelling errors (n) Jaccard, cosine and word2vec cosine similarities between Q and A strings, # of unique and common words in A and Q respectively
Questioner personalization features
of questions Questioner has asked, average number of answers per question, 23 linguistic features from above averaged over the question question answer time, Best Answerer points, rank of BA, % of questions with > 1 answer, aggregated personalization results for best (BA) and non-best answers (NonBA) submitted for each of Q's questions where features are aggregated over time, points and linguistic features (τ), and the ratio of all of the respective BA and NonBA features
Answerer personalization features
average num of answers (τ and log), # of BAs (τ and log), BA percentage average questioner points (log), average rank total number of answers (includes himself and community) to all questions he answered, ratio of total number of answers to above total (henceforth "ratio") answerer quality metric 1: (# of BAs / # of As - ratio) (τ), answerer quality metric 2: best Answer percentage / random baseline (τ) answerer quality metric 3: (ratio - 1) (τ), 23 linguistic features from above averaged for all of answerer's questions
5 Answerer "superlative" features
answerers with 1) most BAs, 2) highest BA percentage, and highest quality metrics

Table 1. Feature Overview. For features which take on numeric values, we use the actual value, in addition to that value expressed as a log (log) or a boolean vector (τ). Linguistic features which are a count of specific words or patterns were in many cases normalized by the number of tokens in the string (n). Some features were derived specifically for the answer(er) A or question(er) Q.