Building a Parsimonious Model for Identifying Best Answers Using Interaction History in Community Q&A

Chirag Shah
School of Communication & Information
Rutgers University
4 Huntington St, New Brunswick, NJ 08901
chirags@rutgers.edu

ABSTRACT
Evaluating answer quality or identifying/predicting which answer would be selected as the best for a given question is an important problem in community-based Q&A services. In this article we introduce new interaction-based features depicting the amount of distinct interactions between an asker and answerer over time, in order to predict whether an answer will be selected as Best Answer or not within Yahoo! Answers. Through a series of experiments ran on a data set of 23,218 question-answer pairs, we determined that after the data was first run using a model trained on textual features, and then the failed cases re-run with a model trained on interaction features, we were able to significantly improve the performance of the original model in identifying these difficult cases. In addition, when compared to models using often five to seven times the amount of features and requiring a large amount of computational effort, our model performed at to above the same evaluative measures. This suggests that future classification models can be made more parsimonious and handle larger datasets using less computational effort by developing a two-step classifier that includes interaction history as a feature.

Keywords
Community Q&A; Online communities; Interaction history; Model building.

INTRODUCTION

Community-based question-answering (CQA) services such as Yahoo! Answers have become an exceedingly popular way for Web searchers to solicit information, advice, and opinions from their peers. CQA represents a Web 2.0 application that capitalizes on collaboration between communities of users for information exchange (Shachaf & Rosenbaum, 2009). Yahoo! Answers represents one of the most popular CQA services with over 100 million users and one billion questions asked (Adamic et al., 2008). Users can interact within Yahoo! Answers not only by asking and answering questions, but also by voting and commenting on answers. In addition, the asker of a question can assign a Best Answer to an answer he or she feels best fulfills his or her information need. Yahoo! Answers content, including these human-based relevance judgments are available via the Application Programming Interface (API).[1]

Although Yahoo! Answers has over one billion questions asked, only a fraction of these questions are resolved, meaning they have an answer that has been chosen as Best Answer by the asker (Shachaf & Rosenbaum, 2009). There are several reasons for studying which answers is picked as the Best Answer and why. For instance, being able to identify a good/best answer could lead to understanding and recommending ways to improve answers within CQA. Given that CQA's success relies heavily on community building and participation as well as askers receiving satisfactory answers, a *healthy* behavior such as constructing a good answer could lead to increased trust, usage, and participation within a CQA. This study therefore follows in this vein of inquiry by creating a classifier that can correctly predict Best Answers within Yahoo! Answers. In order to build a successful classifier, we must identify features that contribute to distinguishing a Best Answer from a Not Best Answer and argue that interaction history between askers and answerers constitutes an important set of features.

Shah, Oh, and Oh (2009) identified three primary components of CQA services: a way to ask a question, a way to answer a question, and the participation around such question-answering (Q&A) activities. As they, and others, recognized – it is the participation around Q&A activities that is the most defining characteristic of such CQA services. Unfortunately, the majority of research conducted in CQA domain with data mining approaches has primarily focused on analyzing the Q&A content, and not the nature

ASIST 2015, November 6-10, 2015, St. Louis, MO, USA.

[1] http://developer.yahoo.com/answers/

of the participants' interactions around this content (Shachaf & Rosenbaum, 2009).

The present paper will therefore focus on mining data for not only Q&A content, but also the interactions among askers and answerers within a CQA. In addition, we hope that by introducing interaction features into a small set of textual features, we can create a parsimonious – a minimalistic – model that predicts answer quality within Yahoo! Answers.

In order to meet these aims, we will use a large amount of data from Yahoo! Answers. Specifically, this can be translated into the following research questions:

RQ1. How similar (in terms of the accuracy) a classifier using a limited amount of features perform to classifiers built using many features?

RQ2. To what extent the amount of distinct interactions between an asker and answerer constitute a predictor of whether an answer will receive a Best Answer rating?

Since the asker and answerers may not know one another, and a typical CQA site is not designed to provide real-time synchronous interactions among them, we will use the asynchronously exchanged messages as a representation of interaction. Interaction, therefore, more specifically is defined here as the amount of times an answerer provides an answer to an asker's question.

To address these research questions, the work reported here will present experiments with a large set of question-answering and interaction data from Yahoo! Answers, the largest CQA service on the Web (Agichtein et al., 2008). Specifically, classification models will be built using different kinds of features, including those with askers' and answerers' interactions to predict the selection of the Best Answer from a set of answers for a question.

The rest of the paper is organized as follows. First, background on CQA in regard to answer classification and quality will be outlined. Then, our methods for obtaining the features for our classifier, as well as the classification algorithms used will be discussed. Findings will then be presented followed by a discussion and conclusion.

BACKGROUND

Given the access to a large database of human relevance judgments approximated by Best Answer ratings and the need to moderate the variable quality of content exchanged within these sites, many information retrieval researchers have attempted to determine the relationship between various features generated from within Yahoo! Answers and human judgments (Agichtein et al., 2009; Shachaf & Rosenbaum, 2009). While the aggregated quality of the total content within CQA has been found reasonable, Su et al. (2007) discovered that within individual cases, the quality of answers is highly variable – 17-45% of the answers provided to questions posed by the authors were correct as opposed to 65-95% of questions rated as Best Answers within a large-scale sample. This indicates that a system for retrieval of answers ranked as high quality based on the similarity of the question asked to those archived in the database could improve the satisfaction of users.

Researchers addressing this problem haven typically created classifiers, which identify various aspects of answer quality, often within Yahoo! Answers by using Best Answer ratings as an indication of answerer satisfaction, thus attesting to the perceived high quality of the answer (e.g., Liu et al., 2008). Several approaches exist to identifying Best Answers. Liu et al. (2008) generalized these approaches into an Asker Satisfaction Prediction (ASP) framework, including textual and semantic features of questions and answers, history of answer satisfaction by category, and past activity history of askers and answerers (Belkin et al., 1982; Jeon et al., 2006; Liu et al., 2008; Agichtein et al., 2008; Shah & Pomerantz, 2010; Shah, Kitzie, & Choi, 2014). Other studies have attempted to both add classification features and use other evaluative baselines for answer quality using human-based assessments. Examples of the former include typologies for question type labeled by human assessors where findings indicate that the distribution of Best Answers significantly varies among these types (Harper et al., 2009; Liu et al., 2008) and examples of the latter include using human assessments as baselines for answer quality (Shah & Pomerantz, 2010; Agichtein et al., 2008). These classifiers often employ either regression based or probabilistic based analyses including Support Vector Machines and Bayesian Networks (Agichtein et al., 2008; Liu et al., 2008). There have also been attempts to classify questions based on their types, such as advice, factual, opinion, and using that to recommend appropriate Q&A service (e.g., Choi, Kitzie, & Shah, 2012).

Adamic et al. (2008) performed a large-scale analysis of Yahoo! Answers, with 8.4 million answers, 1.1 million questions, and 700,000 distinct users. They found a user tends to provide more Best Answer ratings when their rate of participation (e.g. asking, answering, evaluating) is lower (Adamic et al., 2008). This suggests that users who interact more within Yahoo! Answers might evaluate answers differently than those who do not take as much advantage of the community-based elements of the site, a finding also present in studies of other online Q&A communities, including within the UseNet community (Smith, 2012; Fiore et al., 2002). In contrast to these findings, Agichtein et al. (2008) found that Yahoo! Answers tend to adopt multiple roles (e.g. asking and answering questions) and that it was more difficult to distinguish these users into separate categories based on their participation within the site, which the authors attribute to the incentive mechanisms of the site that require users to gain points by answering and/or evaluating content in order to ask questions. This feature of the Yahoo! Answers service might also account for the noted imbalance between

resolved answers, or answers that receive a Best Answer rating, and total answers (Adamic et al., 2008).

Agichtein et al.'s (2008) study performed one of the first large scale studies combining content-based features and network-based features in order to identify quality answers as ranked by human coders within Yahoo! Answers using the ASP framework with 71 features. Findings indicate that models trained on each set of features from the framework perform at a sub-standard level, but the combination of features leads to adequate classification performance, suggesting that each set of features provides independent information that makes a unique contribution to the model (Agichtein et al., 2008).

Our study differs from past works in a few key ways. First, we acknowledge that while effective, many of the methods employed to collect features for predicting answer quality require more time, labor, and/or computational effort. Given that the collective aggregation of answers within CQA provides average to better than average quality (Shachaf & Rosenbaum, 2009), an effective classifier requires training on a large-scale dataset. Therefore, many of the previous works that employ classifiers requiring a large amount of supervision are not practically well suited to automating question quality within CQA. In addition, many of these past works do not consider the relationship between content exchanged and the network properties of CQA services (Bian et al., 2009).

For these reasons, we chose to build a parsimonious classification model using a minimal amount of features. Given the relative effectiveness of textual features in classifying answer quality (see TREC[2] publications for overview) we decided to incorporate several textual features from answers only that were fairly simple to compute (see the following subsection for more information). In addition, we introduce three interaction features, also simple to compute, that address the relationship between the asker and answerer interaction within each specific question-answer instance. This also differs from past models, which rely more on the profile information of the asker and answerer separately. Our main objective is to determine whether a simplistic classifier can be created from a pared down set of textual and interaction based classifiers that performs with more efficiency and similar effectiveness to those less parsimonious classifiers requiring more computational effort. We will now overview details on methods used to build our feature set.

Our data set consisted of over one million questions and five million related answers extracted from Yahoo! Answers using their search APIs between 2007 and 2009. Each question extracted had five related answers. One of these five was the Best Answer and the other four were randomly sampled answers that did not receive a Best Answer rating. Although many questions received more than five answers, the daily limits of the API combined with the time frame we had to collect the data contributed to this decision to randomly sample four other answers along with the Best Answer. Studies have also found that on average a question on Yahoo! Answers receives about 5-6 answers (Shah, Oh, & Oh, 2008).

Obtaining Interaction Features

For each data entry, we received the following information as fields from the API response: question id, answer id, asker id, answerer id, question subject, question content, answer content, rating (5 signified a Best Answer and 0 signified a Not Best Answer), time stamp when the question was posted, time stamp when the answer was posted, and time stamp when the Best Answer was selected. We also added additional fields by calculating delta values between the time stamp when the answer was posted and the time stamp when the question was posted, as well as delta values between the time stamp when the Best Answer was selected and the time stamp when the question was posted.

We then queried this data set to select all instances where the amount of interactions between an asker and answerer were greater than one. To accomplish this, we asked the system to return all count values of 2 or greater where the asker id and answerer id matched within a row. We also added that the answer provided in these instances was selected as Best Answer, by specifying the rating field to equal 5 ($n_{returned}$ = 230,840). We also queried the data set to again return instances where the asker id and answerer id matched within a row more than one time, but this time specified that the answer was not selected as the Best Answer by specifying the rating field to equal 0 ($n_{returned}$ = 11,658). In both instances, we also created another field to our data set that recorded the number of distinct interactions between asker and answerer.

Given that we wanted to run a binary classification for whether or not a question was selected as the Best Answer, we wanted to neutralize our baseline by having an equal amount of Best Answers and Not Best Answers. Therefore, we randomly sampled 11,609 question-answer pairs from those returned instances where there was more than one distinct interaction and the answer was chosen as Best Answer for a total of N=23,218.

The types of interaction features we used for this study were:

- *count_num* Number of distinct interactions between the asker and answerer.
- *Δchosenanswer* Difference between the timestamp of when the answer was chosen as Best Answer by the asker and the timestamp of when the question was posted.
- *Δanswer* Difference between the timestamp of when the answerer's answer was posted and when the question was posted.

[2] http://trec.nist.gov/pubs.html

Obtaining Textual Features

Once we had our data set, we extracted the following textual features from the answer content using a Java extractor we wrote containing the following features:

- *complexwords* A dictionary was created with a list of complex words and the extractor assigned a complexity score to a question based on the presence of these words.
- *readingscore* Flesch-Kincaid Readability scores (Jeon et al., 2006) were calculated to determine the reading ease of a question based on the amount of syllables contained within a word. The higher the score, the more easily understood the piece of content.
- *averagewords* This measure indicates the amount of novel information communicated within the question, which may assist an answerer in interpreting an asker's information need with a higher level of specificity, improving the overall answer quality. This is calculated by counting the number of distinct words over the total number of words used in a question.
- *numquestions* Content containing multiple questions might confuse the answerer in interpreting what information the asker is looking for. A script identified the number of unique question marks in order to assign a related score. Question marks directly adjacent to one another were only counted once.
- *misspelledwords* A dictionary was created that measured misspellings using Jazzy[3], a Java based spell checker built on the Aspell algorithm. Questions containing many misspelled words may be unclear to the reader.
- *entropy* Entropy measures the similarity between the language used within questions asked on Yahoo! Answers and the language used in the baseline comparison exemplar data corpus, in this case the LA Times collection available from TREC[4]. The higher the entropy score, the less clear the question.
- *numcharacters / numwords / numsentences* Often the length of an answer affects its quality (Shah et al., 2009). Sometimes answers might be too short and not provide enough information, while in other cases, answers might be too long and provide superfluous information that ultimately confuses the reader or demands too much of them in answering the question.

Once we collected the answer features, we combined them with the other non-textual (interaction) features for a total of 12 features to be used in our prediction model.

[3] http://jazzy.sourceforge.net/
[4] http://trec.nist.gov/data/docs_eng.html

Overall Classification Framework

Our classification problem is binary, asking the classifier to distinguish Best Answers from Not Best Answers. Given that the objective of the classifier is to identify Best Answers, when findings are presented we will not only look at accuracy as a classification metric, but also recall, precision, and the area under the ROC curve (AUC). These latter three measures approximate the ability of the classifier to discriminate Best Answers from Not Best Answers, while accuracy depicts overall performance.

All classifications were performed using the Weka framework (Witten & Frank, 2005). We experimented with classification algorithms that have been empirically successful in completing classificatory tasks with textual features including support vector machines and logistic regression. We found that a Bayesian Network (BayesNet) classifier gave us the best performance. This algorithm classifies based on the prior probabilities of the occurrence of each feature (Agichtein et al., 2008). Specifically, it takes into account that features within Yahoo! Answers are relational and maps the connection between different elements and their features based on these relations. For this reason, a BayesNet algorithm should therefore enhance the performance of a classifier trained on interaction features that hinge on the interrelationship between asker and answerer (Agichtein et al., 2008). In addition, it is a relatively simple and fast classifier (Liu et al., 2008), which lends to classification using less processing speed. The next section provides details of our experiments and the results.

EXPERIMENTS

Descriptive Statistics

Table 1 indicates descriptive statistics for the full dataset used in the following two subsections.

Table 1: Descriptive Statistics (N=23,128).

	Min	*Max*	*Mean*	*StdDev*
numcharacters	1	25	14.07	7.33
numwords	0	1559	30.95	76.46
averagewords	0	69	0.97	0.63
complexwords	0	410	3.48	14.20
misspelledwords	0	1413	8.63	23.68
taboowords	0	5	0.01	0.10
numsentences	0	326	2.06	5.47
numquestions	0	99	0.31	2.07
readingscore	-1308	122	83.70	37.89
entropy	0	9	3.31	1.77
count_num	2	495	15.83	38.03
Δ chosenanswer	14405	1155327	49270.20	61832.72
Δ answer	13	663199	8829.68	23151.78

Table 2 indicates the distribution of the dataset into 25 top-level Yahoo! Answers categories (the 26th category, Yahoo! Products, had not been created when this data was sampled). Askers must choose one category to label their questions for archival purposes. When compared to other works on question categories within Yahoo! Answers and given that some categories are more popular than others, our distribution of question-answer pairs by question category appears to approximate a random one.

Table 2: Question Category Distribution.

Arts & Humanities	997
Beauty & Style	354
Business & Finance	786
Cars & Transportation	468
Computers & Internet	506
Consumer Electronics	579
Dining Out	789
Education & Reference	710
Entertainment & Music	3,454
Environment	705
Family & Relationships	346
Food & Drink	1,198
Games & Recreation	803
Health	307
Home & Garden	339
Local Businesses	528
News & Events	2,768
Pets	427
Politics & Government	791
Pregnancy & Parenting	391
Science & Mathematics	838
Social Science	363
Society & Culture	692
Sports	2,056
Travel	2,023
Total	*23,218*

Although past research has found that question category within Yahoo! Answers correlates with Best Answer ratings (Adamic et al., 2008), we chose to only include this as a descriptive feature and not incorporate it in the model. We made this decision due to our decision to use interaction history between asker and answerer as a feature in our model. Therefore even if interaction history varies by question category, variation by question category would be redundant in the model since we already have unique interaction history between each asker and answerer pair.

Predicting Best Answers using One-Step Classifier

In the first set of experiments, we evaluated the performance of textual features and interaction features in predicting whether or not an answer receives a Best Answer Rating on our data set of N=23,218 questions. This data set was randomly divided into 70% training (n=15,478) and 30% testing (n=7,740). Our first three experiments indicate the ability of textual features, interaction features, and textual features + interaction features to classify a Best Answer from the test set and are displayed in Table 3.

Table 3: Correctly/Incorrectly Classified Instances, Precision, Recall, and Area Under the Curve for Correctly Classifying Best Answers and Not Best Answers on Training/Testing Set.

	CCI	ICI	P		R		AUC
	%	%	BA	NBA	BA	NBA	
Textual (Baseline)	87.24	12.76	0.96	0.77	0.84	0.94	0.98
Interaction	63.62	36.38	0.78	0.5	0.59	0.71	0.73
Textual + Interaction	89.24	10.76	0.97	0.8	0.86	0.95	0.98

While the interaction model does not perform much better than average on its own, adding interaction features to the baseline model appears to increase the performance of the model in classifying Best Answer instances. In addition, an information gain calculation ran on the data, which considers both the predictive power of each feature and the redundancy between them[5] is displayed in Table 4.

These ratings suggest that the interaction features (in grey) are among the top features ranked as having a combined high predictive power and low redundancy with other features, although compared to the first three features these rankings are relatively low, which might be attributed to redundancy between them, an issue which will be further explored in subsection "Running a Two-Step Classifier on Unseen Data". However, the ranking of interaction features relative to other textual features suggests that the contribution of interaction features to a model trained on textual features should further be explored.

In order to determine whether interaction features make a contribution in classifying Best Answer instances independent from textual features, Krippendorff's Alpha (Hayes & Krippendorff, 2007) was calculated to determine

5 http://weka.sourceforge.net/doc.dev/weka/attributeSelection/package-summary.html

the agreement among error scores for each testing case classified by the model trained on interaction features and the model trained on textual features. The resulting α= 0.199 signifies a low level of agreement. Although it should be noted that the poorer performance of the interaction model adds noise to this score, it appears that there are instances in which the interaction model performed a correct classification where the textual features model did not, and vice versa. For this reason, we decided to examine the performance of interaction features on the failed classification instances of the model trained on textual features.

Table 4: InfoGain Ratings.

Attribute	Score
readingscore	0.949428
entropy	0.883942
averagewords	0.571158
count_num	0.07305
Δchosenanswer	0.044183
Δanswer	0.04341
numwords	0.039532
numcharacters	0.030621
misspelledwords	0.026932
numsentences	0.017472
complexwords	0.017213
numquestions	0.00854
taboowords	0.000973

Predicting Best Answers Using a Two-Step Classifier

In the next set of experiments, we determined the viability of a model in which a two-step classifier was employed. In the first step, instances would be classified using the model trained on textual features. In the next, these failed instances would be re-classified using a model trained on interaction features.

We took the results from the baseline model developed previously and extracted the failed classification instances (n=988). Table 5 depicts the results of classification experiments on these failed instances using the models trained on the training data in subsection "Descriptive Statistics".

The results from these experiments confirm the observation made in subsection "Descriptive Statistics" that models trained on textual versus interaction features correctly classify Best Answer and Not Best Answer instances among different data points depending on the model used. As indicated by the results, the interaction model performs a little better than average in regard to results classified, but when compared to the textual model, these results herald a significant improvement as they classify 59% of those failed instances incorrectly classified by the first classifier. In addition, the level of discrimination of the interaction model is high, as indicated by the area under the ROC curve as depicted in Figure 1. This suggests that while the classifier performs slightly better than normal, it has a higher instance of successes in identifying Best Answers from the data set, which is the intended goal of the classifier.

Table 5: Correctly/Incorrectly Classified Instances, Precision, Recall, and Area Under the Curve for Correctly Classifying Best Answers and Not Best Answers on Failed Classification Set.

	CCI	ICI	P		R		AUC
	%	%	BA	NBA	BA	NBA	
Textual (Baseline)	0	100	0	0	0	0	0
Interaction	59	41	0.893	0.266	0.569	0.696	0.725

The results from these experiments confirm the observation made in subsection "Descriptive Statistics" that models trained on textual versus interaction features correctly classify Best Answer and Not Best Answer instances among different data points depending on the model used. As indicated by the results, the interaction model performs a little better than average in regard to results classified, but when compared to the textual model, these results herald a significant improvement as they classify 59% of those failed instances incorrectly classified by the first classifier. In addition, the level of discrimination of the interaction model is high, as indicated by the area under the ROC curve as depicted in Figure 1. This suggests that while the classifier performs slightly better than normal, it has a higher instance of successes in identifying Best Answers from the data set, which is the intended goal of the classifier.

Running a Two-Step Classifier on Unseen Data

The results of our previous experiments indicated that a model trained on textual features versus one trained on interaction features would correctly classify different instances of a Best Answer within a dataset. Our unseen dataset is comprised of N=54,403 question-answer pairs randomly sampled from the same database in which the other data was acquired. It excluded data previously sampled for past experiments. A key difference in this data was the distributions in the number of interactions, which were lower since we did not require this dataset to have a count of two distinct interactions or more. Given that this dataset was selectively sampled using instances where interactions between asker and answerer were two or more, we wanted to test our models on a more realistic dataset in which the number of interactions between asker and answerer are often only one. As a comparison, Table 6 depicts differences in central tendencies of number of

interactions between the data set used in subsections "Descriptive Statistics" and "Predicting Best Answers using One-Step Classifier" as compared to the unseen dataset.

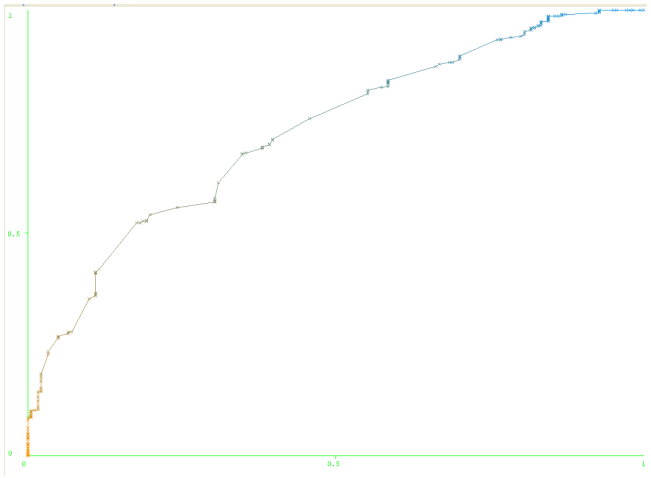

Figure 1: Area Under ROC Curve for Model Trained on Interactions and Tested on Failed Classification Instances (AUC=0.725).

Table 6: Central Tendencies for Both Datasets.

		Min	Max	Mean	StdDev
Training Dataset	n=15,478	2	495	15.833	38.023
Unseen Dataset	n=54,403	0	495	5.032	14.117

Predictably, the results of classifications as depicted by Table 7, which used the models trained on textual features, improved. This should come as no surprise given that interactions were less variable as indicated by the standard deviation of the unseen dataset, which is more than half the standard deviation of the training dataset. Encouragingly, however, the performance of the interaction classifier remained relatively the same, suggesting its consistency in flagging failed classifications from models trained on the textual features.

Table 7: Correctly/Incorrectly Classified Instances, Precision, Recall, and Area Under the Curve for Correctly Classifying Best Answers and Not Best Answers on Training/Testing Set.

	CCI	ICI	P		R		AUC
	%	%	BA	NBA	BA	NBA	
Textual (Baseline)	97.36	2.64	0.98	0.967	0.967	0.981	0.998
Interaction	57.81	42.19	0.635	0.555	0.368	0.788	0.64
Textual + Interaction	97.59	2.41	0.981	0.971	0.97	0.982	0.999

Given this performance, we then attempted the same experiment performed in subsection "Predicting Best Answers using One-Step Classifier", in which we took the failed classifications from the model trained on textual features and classified them using the model trained on textual features and the model trained on interaction features. Table 8 indicates these results.

Table 8: Correctly/Incorrectly Classified Instances, Precision, Recall, and Area Under the Curve for Correctly Classifying Best Answers and Not Best Answers on Failed Classification Set.

	CCI	ICI	P		R		AUC
	%	%	BA	NBA	BA	NBA	
Textual (Baseline)	0	100	0	0	0	0	0
Interaction	60	40	0.759	0.47	0.527	0.696	0.715

The performance of a model trained on the interaction features is similar to the model achieved using training and testing data, which shows the validity of our model on unseen data. Again, the ROC curve denotes the ability of this model to correctly classify Best Answer instances as denoted by Figure 2, although performance has somewhat decreased.

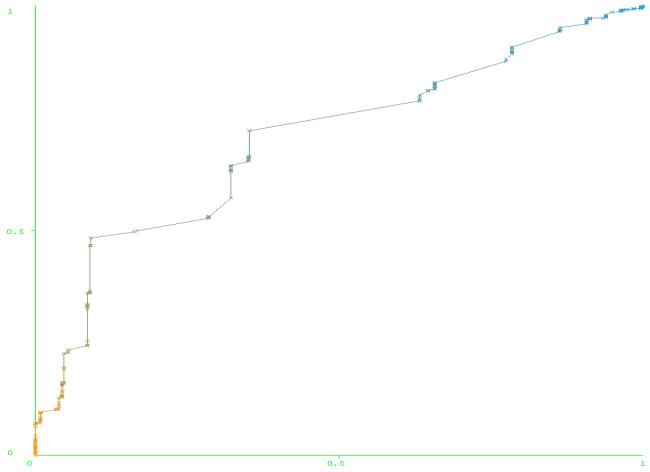

Figure 2: Area Under ROC Curve for Model Trained on Interactions and Tested on Failed Classification Instances (AUC=0.715).

Additional Experiments Using Number of Interactions Only

Table 3 depicted the relative predictive power combined with redundancy of textual features and interaction features. While findings suggested that interaction features outranked many textual features in ratings, the relative low rating in comparison to the first three textual features suggested that either the predictive power of these features was low or they were redundant. Our experiments reported in the earlier subsections indicated that interaction features could play a discriminatory role in correcting failed instances from a classification model built on textual features. However, further experimentation was needed to determine

whether these interaction features were redundant; if so, then perhaps the power of the classifier could be improved.

Table 9: Correlation Attribute Selector.

Attribute	Score
averagewords	0.1423
readingscore	0.1419
acmplexwords	0.1086
Δ answer	0.1071
numcharacters	0.0985
Δ chosenanswer	0.0966
numwords	0.082
numsentences	0.0737
entropy	0.0723
count_num	0.0523
misspelledwords	0.0505
numquestions	0.0371
taboowords	0.0321

Table 9 indicates a correlation attribute table, which measures redundancy, ran on the training set of textual and interaction features. When comparing these results to the InfoGain ratings, it appears that some results are in relatively the same position while others have moved around. The interaction features (highlighted in grey) *Δanswer* and *Δchosenanswer* both have a higher level of redundancy when compared to *count_num*, which indicates the number of distinct interactions. This suggests that perhaps the two time-based features might be contributing noise to the interaction classification model.

For this reason, the failed classifications from the unseen dataset trained on textual features were re-ran using number of distinct interactions as the only predictor. The results are reported in Table 10. Although the overall classification power of the model has increased slightly, the AUC value has slightly decreased, which suggests that a model only trained on number of distinct interactions does not offer significant improvements as compared to when delta time values are incorporated, leaving us to conclude that both interaction counts and time-based features accentuate one another in making a contribution to the interaction-based model.

The Low Processing Cost of Our Classifier

In order to demonstrate the low cost of training-testing data required for our classifier, we decided to use a model trained on 30% of textual + interaction features developed in subsection "Description Statistics" (n=7,440) and test its performance on 70% of the data (n=15,478). Results indicate that even given a small portion of training data, the classifier is able to perform with the same level of accuracy as when trained on 70% of the data as indicated by Table 11.

Table 10: Correctly/Incorrectly Classified Instances, Precision, Recall, and Area Under the Curve for Correctly Classifying Best Answers and Not Best Answers on Failed Classification Set.

	CCI	ICI	P		R		AUC
	%	%	BA	NBA	BA	NBA	
Num distinct interactions	64.7	35.2	0.881	0.283	0.658	0.602	0.636

Table 11: Correctly/Incorrectly Classified Instances, Precision, Recall, and Area Under the Curve for Correctly Classifying Best Answers and Not Best Answers on Training/Testing Set (30/70).

	CCI	ICI	P		R		AUC
			BA	NBA	BA	NBA	
Textual + Interaction	98.89	1.11	0.979	1	0.978	1	0.999

This suggests that our model not only is efficient in using a smaller amount of features requiring a lower computation cost to obtain and allowing us to test on larger datasets, but also that the performance of this model even when trained on a smaller amount of data still correctly classifiers Best Answer instances with remarkable accuracy.

DISCUSSION

This section presents a summary of our experiments and findings, as well as the limitations of our approach.

Summary of Findings

Findings from this study indicate that interaction features make a distinct contribution to a model trained on textual features identifying Best Answers from Yahoo! Answers.

In our first set of experiments, we explored the performance of various models trained on baseline textual features, interaction features, and their combined effects using 70% of the data for training and 30% for testing. When comparing the performance of the model trained on textual features to one trained on textual and interaction features, the accuracy of the model increases. However, an increase in accuracy is not enough to be able to determine whether interaction features make a distinct contribution to the model. For this reason, we compared the difference of performances between the model trained on textual features to the model trained on interaction features, finding significant differences between error rates for each individual prediction. What this suggested is that even though the model trained on interaction features did not perform well on its own, it was able to correctly classify instances of classification errors made by the model trained

on textual features. Specifically, the model trained on interaction features performed well in correctly classifying false negatives made by the classifier.

We then took the failed classification instances made by the classifier trained on textual features and classified these failed instances using the model trained on interaction features. Although the classification accuracy of the model performed slightly better than average, the area under the ROC curve indicated the ability of the model to correctly identify Best Answers, which was a weaker area in the model trained on textual features. For comparison, a model trained on textual features was re-ran on these failed instances to indicate that the baseline performance of this classifier performed at 0% accuracy. Given this performance, our interaction-base classifier on failed instances improves the classification of failed cases on the original model by over 50%. These findings are encouraging as they indicate that interaction features make a numerically small, yet unique contribution to correctly classifying Best Answers.

In order to demonstrate the validity of this model, we then re-ran our training models on unseen data. We found that the performance of the text-based classifier greatly improved on this data, most likely because data sampled was not limited to instances in which the asker and answerer interacted twice or more. At the same time, the performance of the classifier trained on interaction features remained better than average, which suggests that even when the distribution of interactions is normalized and skewed even more greatly toward the right, a model trained on interaction-based features still has the viability to make a distinct contribution.

Therefore, we again ran the failed classification instances from the first model trained on textual features on the model trained on interaction features, again finding that the classification performance increased by more than 50%. In addition, further experiments revealed that the combined power of the interaction features performed best in classifying these failed instances.

While these findings are experimental, they ultimately suggest a research avenue for classification within CQA that incorporates more of the community-based features within the site. Although previous studies have addressed this by using metadata, we have (1) introduced three new interaction features that should further be explored in more expansive models, and (2) highlighted their distinct contribution in fine-tuning the performance of a classifier trained on textual features. Further study could be done to provide a comprehensive identification of community-based interactions within CQA and measure how they interact with textual-based features in order to derive more effective and efficient models.

Limitations

One of the largest limitations of this study was a lack of additional metadata and other features used in previous models. The reason why we did not incorporate these features into this model is due to the exploratory nature of this study; in particular, we wanted a manageable amount of features in order to better understand the relationship between a very popular form of classification within CQA, textual features, and a less studied form, interactions among the asker and answerer. Furthermore, the performance of our classifier demonstrates the possibility to have a model that performs effectively and economically, both by identifying with relatively high accuracy Best Answers and then fine-tuning these results using a second classifier and also by being efficient, particularly since dealing with many features often can adversely affect computational costs.

Another limitation is selected of Best Answer as a metric for answer quality given the small proportion of answers selected as Best Answers within the Yahoo! Answers database (Liu et al., 2008). As Liu et al. (2008) argue, this indicates a larger problem with asker dissatisfaction that should be addressed using other evaluative baselines. One solution for future work can be to further experiment using human assessments. For example, Shah & Pomerantz (2010) used Amazon's Mechanical Turk (MTurk) workers to evaluate answer quality along 13 relevance criteria, and determined that these assessments were internally consistent and approximated expert-based quality ratings. Further tests using the interaction features we have introduced in order to classify answer quality could lend further insight into whether interaction history precludes answer quality or vice versa.

CONCLUSION

In this paper, we attempted to build a classifier that could identify an asker-assigned Best Answer within the CQA service, Yahoo! Answers. Our classifier differs from classifiers built in previous studies in a few key ways. First, we used a limited number of features (12) divided into two feature sets: textual features and interaction features. In both cases, the features were facile to collect and required less computational processing than features used in other studies, which tended to use many more features (in the literature we reviewed, often five to seven times the amount of features than used in this study). We also based our evaluative metric, assessments from Yahoo! Answers users, around the idea that this assessment data was easy to collect and did not require any additional human-based assessments or labeling. In addition, assessments from Yahoo! Answers members indicate quality judgments made by the community of users in which our classifier would be created to serve. This saved time and effort of both computation and human-based processing allowed us to experiment with a larger dataset than usual, N=23,218 question-answer pairs used for training-testing instances and N=54,403 question-answer pairs used as unseen testing data.

We then used a probabilistic classifier, BayesNet, to solve the binary classification problem of identifying Best Answers. We found that the classifier trained on both sets

of features (interaction + textual) performed best on training (N=15,778) and testing (N=7,440) based experiments. Further, we identified that the incorporation of these interaction features made a distinct contribution to a model trained on textual features by demonstrating the ability of a model trained only on interaction features to distinguish Best Answers from Not Best Answers on failed cases resulting from a model built on textual features only (AUC=0.725). Testing on an unseen dataset (N=54,403) yielded similar results. We also tested our three interaction features for redundancy and found that each feature makes a distinct contribution to the model.

In order to further highlight the low processing cost of our classifier, we then randomized our first dataset (N=23,218) and switched the distribution of training (N=7,440) and testing (N=15,778) sets in order to show that the classifier performed similarly using a much smaller training set.

What these experiments indicate is that when completing classification problems within CQA, although the classification power of a model may improve with more features that are diverse, the processing speed and computing costs of running these models on the large datasets required renders them not practical for actual use. We therefore developed a model comprised of textual features that can be effectively run on large datasets without high computational cost, and also incorporated a set of interaction-based features that improve its effectiveness.

ACKNOWLEDGEMENT

The author is grateful to Erik Choi and Vanessa Kitzie for their contributions that led to some of the initial analyses of the results. The work reported here is partially funded by the National Science Foundation (NSF) BCC-SBE award no. 1244704.

REFERENCES

Adamic, L. A., Zhang, J., Bakshy, E., and Ackerman, M. S. (2008). Knowledge sharing and Yahoo! Answers: Everyone knows something. In *Proceedings of the International World Wide Web (WWW) Conference*. Beijing, ACM.

Agichtein, E., Castillo, C., Donato, D., Gionis, A., & Mishne, G. (2008). Finding high-quality content in social media. *In Proceedings of of ACM Web Search and Data Mining (WSDM) Conference*.

Bian, J., Liu, Y., Zhou, D., Agichtein, E., & Zha, H. (2009). Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the WWW* 2009: 51-60

Belkin, N., Oddy, R. N. & Brooks, H. M. (1982). Information retrieval: Part ii. results of a design study. *Journal of Documentation*, *38*(3):145–164.

Choi, E., Kitzie, V., & Shah, C. (2012). Developing a typology of online Q&A models and recommending the right model for each question type. *Proceedings of the American Society for Information Science and Technology*, *49*(1), 1–4.

Fiore, A.T., LeeTiernan, S., & Smith, M.A. (2002). Observed behavior and perceived value in Usenet newsgroups: Bridging the gap. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (ACM CHI) (pp. 323–330). New York: ACM Press.

Harper, F.M., Moy, D. and Konstan, J. (2009). Facts or friends? Distinguishing informational and conversational questions in social Q&A sites. In *Proceedings of the 27th ACM Conference on Human Factors in Computing Systems (ACM CHI)*.

Hayes, A.F. & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures* 1*(1)*, 77-89.

Jeon, J., Croft, B., Lee, J.-H., & Park, S. (2006). A framework to predict the quality of answers with non-textual features. In *Proceedings of ACM SIGIR 2006* (pp. 228–235). New York: ACM Press.

Kincaid, J. P., Fishburn, R. P., Rogers, R. L. & Chissom, B. S. (1975). Derivation of new readability formulas for navy enlisted personnel. Technical Report Research Branch Report 8-75, Millington, Tenn, Naval Air Station.

Liu, Y., Bian, J., & Agichtein, E. (2008). Predicting information seeker satisfaction in community question answering. Proceedings of the *ACM SIGIR Conference.* Singapore.

Shachaf, P., & Rosenbaum, H. (2009). Online social reference: A research agenda through a STIN framework. *Proceedings of iConference 2009*, Chapel Hill, NC.

Shah, C., Oh, J. S., & Oh, S. (2008). Exploring characteristics and effects of user participation in online social Q&A sites. *First Monday*, *13*(9).

Shah, C., Oh, S., & Oh, J. S. (2009). Research agenda for social Q&A. *Library & Information Science Research*, *31*(4), 205–209.

Shah, C. & Pomerantz, J. (2010). Evaluating and predicting answer quality in community QA. *Proceedings of ACM SIGIR 2010 Conference*. Geneva, Switzerland: July 19-23, 2010.

Shah, C., Kitzie, V., & Choi, E. (2014). *Questioning the question - Addressing the answerability of questions in community question-answering*. Proceedings of Hawaii International Conference on System Sciences (HICSS), Waikoloa, HI.

Smith, M. (2002). Tools for navigating large social cyberspaces. *Communications of the ACM*, 45(4), 51–55.

Su, Q., Pavlov, D., Chow, J.-H., & Baker, W. C. (2007). Internet-scale collection of human-reviewed data. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, 231-240.

Witten, I. & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. New York, NY: Morgan Kaufman (2nd ed.).