Trust, but Verify: Predicting Contribution Quality for Knowledge Base Construction and Curation

Chun How Tan
Google
chunhowt@google.com

Eugene Agichtein
Emory University, Google
eugene@mathcs.emory.edu

Panos Ipeirotis
New York University, Google
panos@stern.nyu.edu

Evgeniy Gabrilovich
Google
gabr@google.com

ABSTRACT

The largest publicly available knowledge repositories, such as Wikipedia and Freebase, owe their existence and growth to volunteer contributors around the globe. While the majority of contributions are correct, errors can still creep in, due to editors' carelessness, misunderstanding of the schema, malice, or even lack of accepted ground truth. If left undetected, inaccuracies often degrade the experience of users and the performance of applications that rely on these knowledge repositories. We present a new method, CQUAL, for automatically predicting the quality of contributions submitted to a knowledge base. Significantly expanding upon previous work, our method holistically exploits a variety of signals, including the user's domains of expertise as reflected in her prior contribution history, and the historical accuracy rates of different types of facts. In a large-scale human evaluation, our method exhibits precision of 91% at 80% recall. Our model verifies whether a contribution is correct immediately after it is submitted, significantly alleviating the need for post-submission human reviewing.

Keywords

Crowdsourcing, knowledge base construction, predicting contribution quality

Categories and Subject Descriptors

H2.8 [Database Management]: Database Applications

1. INTRODUCTION

The last decade has witnessed an unprecedented growth of publicly available knowledge repositories such as the Open Directory, Wikipedia, and Freebase (and incidentally other repositories that build upon these, including YAGO, DBpedia, and Google's Knowledge Graph). These knowledge repositories thrive thanks to millions of volunteer contributors, who add new information and keep the existing knowledge up to date. All of these repositories employ the post-moderation approach, where the contributions go live immediately, but can later be edited or reverted by other users. This approach has obvious benefits, as it facilitates rapid dissemination of updated information and allows legitimate contributors to get immediate gratification from seeing their contributions online. Alas, this approach is also prone to spamming. In the largest communities, such as Wikipedia, changes on virtually any topic are often reviewed promptly [33]. However, in smaller user communities, such as Freebase, inaccurate contributions can persist for much longer periods of time until they are detected and fixed. Such errors can have significant consequences because these knowledge repositories (notably, Freebase) often serve as data sources for third-party applications, such as Google's Knowledge Graph, Bing's Satori, and Facebook's Entity Graph. Hence, it is important to maintain the knowledge repository at high accuracy level.

There are various ways to maintain high levels of accuracy. One option is to use pre-moderation, which requires all changes to be approved prior to being allowed to go live. Unfortunately, pre-moderation removes the incentive of immediate gratification and negatively affects users' propensity to contribute: in smaller communities facts can remain pending for extended periods of time until they are reviewed and approved. As an alternative to pre-moderation, new submissions can go live immediately, but be considered unverified until they stand the test of time; then, if they have not been deleted, by moderators or other users, after a number of weeks they are assumed to be correct and are released to third party applications. Both options have their shortcomings: the former allows introduction of erroneous facts, while the latter delays the inclusion of valid facts into the knowledge base.

We propose an automatic moderation technique for verifying users' contributions in real time. This approach allows a large fraction of contributions to be automatically approved instantaneously, without the need to subject them to a secondary review or let them stand the test of time. Several earlier studies have examined the quality of Wikipedia contributions based on the quality of contributor's prior work (e.g., [2, 17, 27]), however, we show that these findings do not necessarily apply in the case when contributions are *structured*, such as SPO triples[1] in Freebase. In this case,

WSDM'14, February 24–28, 2014, New York, New York, USA.

http://dx.doi.org/10.1145/2556195.2556227.

[1]SPO stands for a triple of $\langle subject, predicate, object \rangle$, where two entities are connected by a relation predicate.

we observed that naively using the contributors' history to compute their prior accuracy rates, yields results that are barely above the majority class baseline. Instead of using just the past rate of correctness in one's contribution history, we treat the problem of classifying user contributions holistically. We model the contributors' domain of expertise based on their past contributions, as well as the inherent "difficulty" in correctly contributing different types of information. To this end, we use a very large-scale unsupervised topic model to align the users' domains of expertise with the topics of their contributions. To infer the inherent difficulty of a contribution, we track long-term accuracy rates of contributions of different types of facts (and thus compute the prior likelihood of a contribution to be correct).

We conduct an empirical evaluation using a subset of Freebase contributions, which have been re-judged for accuracy by multiple judges to ascertain their quality. Our findings indicate that the prior correctness rate of a user (inferred from his contribution history) only has a slight predictive power with respect to the quality of his future contributions, and the prediction accuracy based on this information alone is not significantly better than that of a simple baseline that always predicts the majority class. However, we measured a significant improvement due to modeling the difficulty of facts, and a further improvement due to modeling the contributor's expertise based on her past contributions.

The contributions of this paper are fourfold. First, we propose an approach for *targeted crowdsourcing*, where we predict the quality of users' work by inferring their *domain expertise* through their past contribution history. Our approach offers a paradigm shift compared to conventional crowdsourcing techniques (e.g., the Amazon Mechanical Turk), where little prior information is available about the users, and qualification tests (even paired with their prior contribution history, if any) shed little light on the quality of their subsequent contributions. Second, our method can do well in the case of a "cold start" when the user has made no prior contributions to the knowledge base, by falling back to the prior difficulty of the type of information being contributed. Third, our empirical evaluation shows that the proposed method can instantaneously vet a very large portion (at least 80%) of the users' contributions, while offering a significant reduction in error compared to the simple test of time heuristic, as well as to the previous state of the art based on users' prior success rates. Finally, as opposed to quality control through redundancy, or via having all contributions explicitly reviewed by humans, or by letting the new contributions stand the test of time, our method offers a continuous tradeoff between the prediction accuracy and the fraction of contributions that can be approved automatically.

2. METHODOLOGY

In Section 2.1, we illustrate the contribution process in Freebase, a popular knowledge base, and formally define the problem of evaluating contribution quality. We then describe the signals used for estimating contribution quality in Section 2.2.

2.1 Life of a Freebase Contribution

Freebase[2] is one of the most popular knowledge bases (as evident by its use by major commercial search engines such as Google and Bing) and serves as a concrete case study in the remainder of the paper. Freebase is primarily updated and expanded by volunteers and "data enthusiasts." User contributions are immediately visible, but may not be *promoted* (i.e., considered correct) until some time later. In Freebase, as in many other knowledge bases (e.g., DBpedia [7]), facts are stored as triples of the form $\langle subject, predicate, object \rangle$.

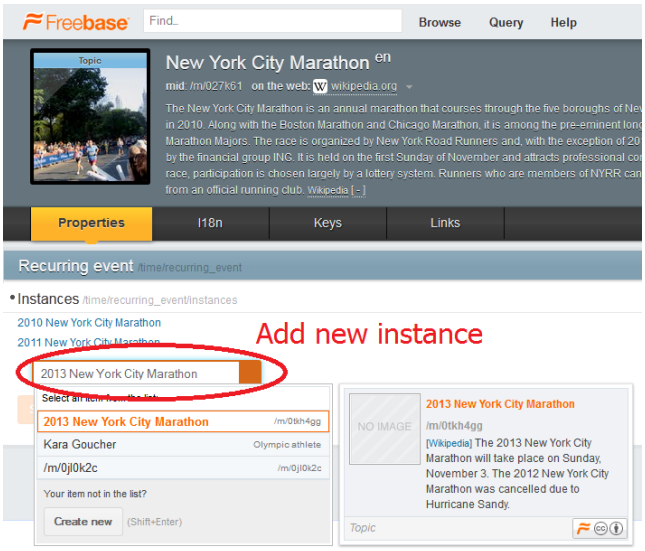


Figure 1: An example Freebase triple contribution screen, where the user adds a new valid instance of a recurring event, namely the 2013 New York City Marathon.

It should be emphasized, however, that this paper does not make any assumptions about the nature of the contributions that are peculiar to Freebase, beyond assuming the information is stored as triples (see the problem statement below). Consequently, the methodology proposed herein can be applied to verify the validity of contributed facts in other similarly-structured knowledge repositories. In our future work, we plan to extend our methodology to judging contributions with a more elaborate structure, such as SPOTL tuples used in YAGO [18], where each SPO triple is further annotated with Time and Location.

The example in Figure 1 illustrates the process of contributing a new triple to Freebase. A user can contribute a new triple, choosing from a rich schema of many possible predicates. For example, a user may creare a new instance of a type (as illustrated in the figure, adding the 2013 instance of the annual New York City Marathon event), or new attributes of an entity (e.g., the date of the event). While most of these contributions are valid, errors occur. Common types of errors include schema errors such as choosing a wrong type for an entity, or a wrong predicate for a relationship, entity reconciliation issues, incorrect facts, and so on. Over time, changes to the knowledge base may be reviewed, and facts can be updated or deleted (or re-added, in a case of deletion). Unfortunately, this process is time-consuming, labor-intensive, and, with existing techniques, tedious for the human reviewers. For instance, Figure 2 illustrates removing an incorrect fact for the birthplace of actor Bruce Lee, which was erroneously stated as "Chinatown," which remained unnoticed for some time, whereas the correct value

[2] http://www.freebase.com/

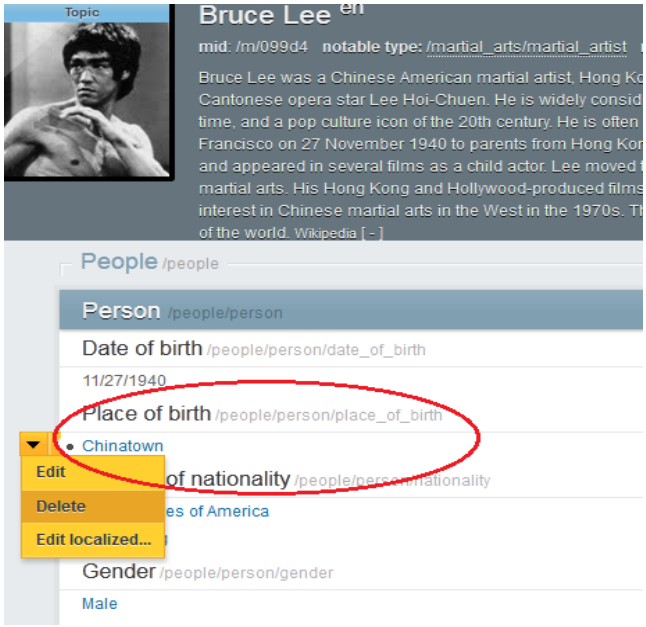


Figure 2: An example deletion of a Freebase triple: removing incorrect birthplace value "Chinatown."

should have been "Hong Kong." Thus, it is possible that erroneous contributions may remain unnoticed for some time, or, conversely, that good (and important) database additions may not be promoted or used by applications until they are reviewed or until they pass the test of time. For example, the contribution related to the 2013 NYC Marathon in Figure 1 is time-sensitive, since the event is scheduled to occur soon, and hence would be an ideal candidate for timely verification. Our aim is to automatically vet the contributions for correctness as soon as they are submitted. Formally, we state the problem as follows:

PROBLEM STATEMENT 1. We are given a contribution of a triple $t = \langle s, p, o \rangle$ by a user u. The goal is to predict whether t is a correct contribution (as judged later by a human review process). The problem is thus formulated as a standard supervised binary classification. □

2.2 Contribution Quality Signals

Our approach is based on the idea that not all users are equally reliable, and that some predicates are inherently more difficult than others. These differences can be captured by various signals that correlate with the validity of the contribution. The signals we consider include user contribution history, user contribution expertise, and triple features (as a proxy for historical predicate difficulty).

2.2.1 *User Contribution History*

For each contribution submitted to Freebase, we use the prior contribution history of the contributing *user*, to estimate the validity of the contribution. Of all the sets of signals that we use, this one is the closest to previous work on estimating users' reputation (e.g., [2, 27, 22]), and we adapt the ideas from that line of research as a state of the art baseline. Specifically, we characterize user's prior contribution history using the features shown in Table 1. It should be noted that when modeling the user's domains of expertise

Feature Description	*Scaling*
Total number of prior contributions	log
Total number of prior *correct* contributions	log
Total number of prior *incorrect* contributions	log
Fraction of correct contribution	–
Total number of deleted (*possibly* incorrect) contributions	log
Fraction of contributions that are deleted	–
Total number of prior deletion actions	log
Number of bad (reverted) deletions	–
Fraction of bad deletions	–
Membership lifetime of user (in days)	–
Number of seconds since last user action	log

Table 1: User contribution history features

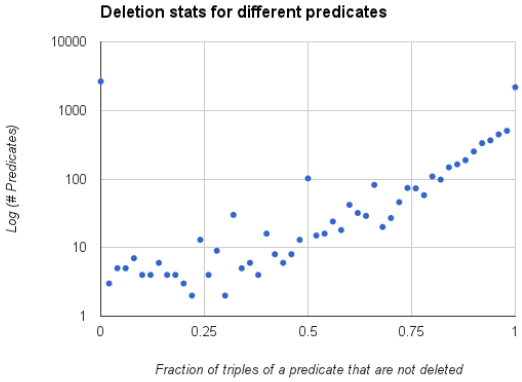


Figure 3: Historical deletion statistics of Freebase predicates

(cf. Section 2.2.3) we also use the set of the user's prior contributions. However, to be compatible with prior work, we only use the term "contribution history" to collectively refer to the features described in Table 1, which characterize the number and the correctness rate of the user's prior contributions.

2.2.2 *Triple Features*

Figure 3 reports the historical deletion statistics of Freebase triples, contributed by the community, for different predicates [3]. The horizontal axis is a proxy for "predicate difficulty": it is the fraction of triples, with a given predicate, that remain undeleted by the community, and are hence considered correct. A high value means that the predicate is "easy" while a low value indicates that the predicate is "difficult." The vertical axis reports the number of predicates in the Freebase schema, that have this "difficulty" value, as computed over all triple submissions with this predicate.

For example, there is a large number of predicates that historically do not contain correct values when contributed by public users (as opposed to Freebase staff, who are more knowledgeable and do not normally spam the knowledge

[3]Freebase data available at https://developers.google.com/freebase/data

Feature Name	*Dimensionality*
Topics	$\sim 1,000,000$
Taxonomy	$\sim 5,000$
Predicates	$\sim 3,000$

Table 2: Representation spaces of contributor expertise.

base). For example, all 120 attribute instances of the predicate /martial_arts/martial_art/well_known_practitioner, which is supposed to contain references to well-known martial artists, submitted by public users, have been deleted by the community; examples of the deleted values include names such as "This is Elvis" and "Theseus", suggesting a strong prior that new values contributed for this predicate are likely to be incorrect. In contrast, values submitted for the predicate "/biology/genomic_locus/strand" are consistently preserved by the community (not deleted), providing a strong prior on accuracy. Finally, a number of predicates such as "/medicine/medical_treatment/side_effects", have roughly equal chances of being correct or incorrect.

As different predicates exhibit different historical deletion rate, the classifier can use this information as a proxy for estimating the prior likelihood of the predicate to be contributed correctly. We capture this information as a vector of features, each corresponding to a fully-specified predicate.

Generalization over predicates.

In Freebase, each predicate is of the form of "/Domain/Type/Property". Examples of top-level domains include "/biology" and "/music". For each domain, there are types associated with it such as "/music/album" and "/music/artist" for the "/music" domain. Finally, for each type, there are properties associated with it such as "/music/album/artist" and "/music/album/genre" for the "/music/album" type. In an attempt to generalize over leaf predicates, we have experimented with introducing additional features corresponding to these domains and types, but found these additional features not to be helpful. One possible explanation is that the information reflected in these features is implicitly captured through the plurality of the leaf-level features, which results in double counting. Experimenting with further refining the prior triple difficulty could be a subject of future refinements of our work.

2.2.3 *User Contribution Expertise*

For each user, we estimate the areas of her expertise based on her previous contributions. To represent the contribution expertise of a user, we use three different concept spaces:

- *Topics*: a large topic model trained in an unsupervised manner using a large web corpus. The *Topics* space is derived from a proprietary scalable implementation of topic modeling very similar to LDA [8], with approximately one million topics.[4]
- *Taxonomy*: This space is based on an in-house taxonomy of approximately three thousand commercial topics, arranged in a four-level hierarchy, and a hierarchical text classifier [23, 13, 26] built in a supervised manner, which classifies text fragments onto the nodes of that taxonomy.
- *Predicates*: the union of all domains, types, and leaf predicates in Freebase (cf. the last paragraph in Section 2.2.2).

[4]More detailed description available at http://www.ipam.ucla.edu/abstract.aspx?tid=10734.

Similarity Metric	*Space*	*Feature Sets*
Dot Product	All	All
Cosine Similarity	All	All
Num. of intersecting concepts	All	Positive, Negative
Jaccard Index	All	Positive, Negative

Table 3: Similarity Metrics for User Contribution Expertise Features

To represent *user contributions*, we map from the triple contribution to the concept spaces described above. The mapping to the Freebase predicate concept space is direct, based on the predicate of the submitted triple. For the other two concept spaces, we use the text properties, and Wikipedia page of the subject and object of the triple, to construct a pseudo-document that contains the textual representation of the subject-object pair. We then process this pseudo-document using the topic model and the taxonomy classifier.

Then, to represent each *user*, we aggregate the Topic-, Taxonomy- and Predicate-representations of their prior contributions. Finally, to model the actual *expertise* of the user, we derive three feature sets based on the representations above:

- The *Positive* set, which aggregates contributions judged to be valid.
- The *Negative* set, which aggregates bad contributions.
- The *Net* set, which aggregates all contributions and is computed as the difference of the corresponding features in *Positive* and *Negative*.

That is, we explicitly model the user's expertise in each domain (represented by the different concept spaces), by tracking the aggregated valid and invalid contributions within each domain.

When a new triple is contributed, we first convert the contribution into the concept spaces described above. Then, to estimate the expertise of the user for each triple, we derive similarity features between the contribution concept distribution and the user contribution expertise distribution. Specifically, we use the similarity metrics in Table 3. Note that we compute similarity metrics involving the intersection count only for the *Positive* and *Negative* sets, but not for the *Net* set, as we just want to capture how similar is the current contribution to the previously proposed good and bad contributions.

While these results are not reported in the paper, during development we examined which of the concept space and expertise representation are most useful. Our analysis suggests that the *Taxonomy* and the *Predicates* concept spaces are more useful than the large *Topics* concept space. This is because the *Topics* concept space has of order of millions of topics, thus spreading the expertise distribution too sparse for users contributing not a lot of triples. This is especially true because the user contributions exhibit the familiar long-tail

distribution, where majority of the users contribute less than hundred contributions and a small number of power users contributing more than hundred of thousands of contributions. In future works, we plan to consider smoothing/smearing the *Topics* space to reduce the number of topics and use other concept spaces. As for the various expertise representation, we observe no significant benefit of using one or the other, but the combination of all of them provide the best performance. This agrees with our intuition since each of the expertise representation intends to cover different facets of the user, i.e., the good, the bad, and the whole.

3. EXPERIMENTAL SETUP

We now describe our methodology for comparing the different methods for predicting the quality of the user contributions on Freebase.

3.1 Datasets

All experiments were performed on Freebase data, a publicly available resource. Each Freebase triple is associated with a timestamp, making it natural to split the data by time. This also allows for easy simulation of an operational environment, where we predict quality of new contributions using only historical data. We focus on contributions from *non-whitelisted users*, i.e., the majority of "regular" users in Freebase, as opposed to white-listed users who are Freebase-approved experts that "own" a particular topic and have the final say on the data therein. The non-whitelisted users have contributed almost 8 million triples to Freebase, so far. We split the data by time, into disjoint training and test sets, as described below.

Test set, using human expert labels: The test set is randomly sampled from the triples contributed by non-whitelisted users between June 1st, 2013 and August 15th, 2013. In total, we obtained judgments from professionally trained human experts for 3,975 triples [5].

Our human expert consists of paid contributors who are trained to do judgment on Freebase and have a deep understanding of Freebase schema. For example, our contributor can deal with intricacies of Freebase schema such as the difference between /people/person and /fictional_universe /fictional_character. All triples were judged by two experts. In case of a disagreement, a third expert adjudicated after a discussion with the original judges. As a result of this procedure, 3,414 contributions were judged as "good," and 561 as "bad."

Training set, using heuristic, test-of-time labels: In order to make use of the millions of historical contributions for training, we had to resort to automatic, *test-of-time heuristic labeling* (it would simply not be practical to manually label a dataset of this size). Using this approach, we assumed that all contributions that survived the "test of time" (i.e., were not deleted) for over K weeks were correct. This heuristic borrows ideas from research on Wikipedia's edit history [2]. Specifically, for heuristic labeling we used the following criteria:

- All triples contributed and survive for older than K weeks that are not deleted are considered good.
- All triples that have been deleted are considered bad unless the triple is reasserted and stays for more than K weeks.
- The act of deleting an contribution is considered good if and only if nobody reassert the same triple that survives for more than K weeks.
- Everything else is labeled as unknown and dropped from training set.

[5]Judgments can be found at http://goo.gl/0CXLYs

Contribution age in weeks	*Labeling accuracy*
2...3	86.5%
3...4	87.5%
4...5	89.3%
5...6	88.8%
6...7	88.4%

Table 4: Testing different cutoff values for the heuristic labeling of training examples.

In order to choose a reasonable value of K, we experimented with different cutoff levels for how long a triple needs to "survive" in order to be considered correct. We sampled about 200 triples submitted in each one week intervals, for different K, (e.g., submitted less than $k+1$ but more than k weeks prior) and obtained manual judgments for them using human experts similar to how we derive our test set. Note that this human-judged data set is disjoint from the test set, and was solely used for development experiments. Table 4 reports the resulting accuracy for different values of K. As we can see, the accuracy increases initially, as we increase K, and then flattens. Consequently, we chose $K = 4$ as a default value for the rest of the experiments in this paper.[6]

The generated training set consists of all triples in Freebase, contributed before June 1st, 2013. We assign labels following the the heuristic "test-of-time" criteria listed above. In total, we have 7,626,924 triples contributed by non-whitelisted Freebase users. We labeled 7,280,900 triples as "good" and 346,024 triples as "bad". The resulting training and test datasets are highly skewed towards "good" contributions (95.46% for training set and 85.9% for test set). The training set is more skewed towards "good" contributions because there might be contributions existing since the beginning of Freebase that are never corrected or that are much higher quality. This is consistent with the reputation of Freebase for being a generally high-quality source of data. Nevertheless, for many applications that rely on this data, accuracy of 86% may not be sufficient, indicating the need for validation or moderation for new contributions.

3.2 Methods Compared

We now summarize the different methods that we compare empirically in Section 4:

[6]We note that the heuristic labels were only used for creating the *training set* (in other words, we trained on data that could be noisy, due to the heuristic nature of the labeling). The experimental performance results are all reported on the test set, which contains only triples that have been judged for correctness by human experts.

- **CQUAL**:[7] Our method, which uses all three feature groups to estimate contribution quality, as described in Section 2.
- **Majority Class, Baseline**: Almost 86% of Freebase contributions are "good" (Section 3). Hence, we have a strong class imbalance between positive and negative examples. In this situation, it is a common practice to use a majority class predictor as a baseline, which predicts every contribution to be correct.
- **Contrib History, current state-of-the-art**: Adaptation of state-of-the-art approaches that predict quality of user contributions are using the prior contribution of the user [2, 27]. We build a supervised classifier that uses only the user contribution history features (Section 2.2.1) and we use it as a more advanced baseline to compare against.
- **Test of Time (Four Weeks)**: Unlike the first two methods, which can be used in real-time, to predict contribution quality as soon as they appear, the "test of time" method can only be used retroactively. This method considers the triples that survived four weeks of Freebase users' scrutiny, to be correct. This could be the moderation approach used in practice, say, by an application that uses data from Freebase: anything contribution submitted less than four weeks ago is not considered part of Freebase. While not truly comparable to the other methods, comparing the accuracy of our predictions to this approach provides a valuable reference point.

3.3 Classification Algorithms

We experimented with commonly used classifiers that could scale to the large number of examples and features.

- *Logistic Regression*
- *Gradboost*: The GradBoost algorithm by Duchi et al., [12], is a generalization of gradient-based coordinate descent methods, shown to be effective for multiclass prediction.
- *Perceptron*: Freund et al. [15].

3.4 Evaluation metrics

We use standard information retrieval metrics of *Precision* and *Recall*, defined respectively as fraction of predicted "good" triples that are truly correct, and the fraction of all true "good" triples identified. Precision is useful to report the rate of true positives, whereas recall measures the coverage of the "good" contributions identified by our method. We also compute the standard classification metrics, namely the ROC curves and the area under the curve (AUC), appropriate for evaluation of classification methods when the class distribution is skewed, as in our setting. We also measure *Relative Error Reduction* (RER), defined as:

$$RER = \frac{error_{baseline} - error_{\text{CQUAL}}}{error_{baseline}} \cdot 100\%.$$

The RER metric is useful to put the differences between system performance levels in perspective, by understanding how much errors we can reduce in our knowledge base using different method given the same recall level.

[7]CQUAL stands for Contribution QUALity predictor.

Method	*RER@25%*	*RER@50%*	*RER@75%*
Majority	0%	0%	0%
Contr. History	44%	10%	0%
Test of Time	22%	22%	22%
CQUAL	**65%**	**44%**	**39%**

Table 5: Relative Error Reduction (RER) at 25%, 50%, and 75% Recall levels.

Method	*AUC*
Majority	0.5
Contribution History	0.543
Test of Time	0.5
CQUAL	**0.707**

Table 6: The AUC values for the methods compared.

4. EXPERIMENTAL RESULTS

In this section we empirically evaluate our methods for predicting the quality of the user contributions on Freebase. First, we present the main results, comparing our approach to various baselines (Section 4.1). We then analyze the results in more depth, to compare the performance of varying the underlying classification algorithms (Section 4.2) and feature sets (Section 4.3). Finally, we conduct an error analysis to provide insights into performance of our approach and possible future improvements (Section 4.4).

4.1 Predicting Contribution Quality

Figure 4 compares the performance of CQUAL to that of several baselines.[8] For each method, we report the interpolated precision values [34] (Y-axis) at each recall level (X-axis). We also computed, at each recall level, a paired t-test, to examine the statistical significance of the differences in performance between the different algorithms. We found that CQUAL outperforms all the baselines, including the retroactive "test of time" method, by a large margin. Specifically, we found CQUAL performance to be significantly superior to all other algorithms at recall levels ranging from 0.25 to 0.85, with the differences being significant at $p < 0.001$.

Table 5 reports the relative error reduction (RER) metric at key recall cut-offs, namely 25%, 50% and 75%. Table 6 lists the AUC values for the different algorithms. CQUAL provides consistent and substantial error reduction at all Recall levels considered, from 65% to 39% relative to the majority baseline, and a significantly higher AUC of 0.707 than competing methods.

4.2 Using different learning algorithms

We also experimented with three different learning algorithms (using all features), namely, logistic regression, gradient boosted log-linear model, and perceptron. Figures 5(a) and 5(b) show the results. Again, all three classifiers yield a substantial improvement over the existing baselines, con-

[8]In this section, we report the performance of the classifier trained using a logistic regression and using all the features described in Section 2.2. We compare performance of different classification algorithms in Section 4.2, and explore the utility of different feature classes in Section 4.3.

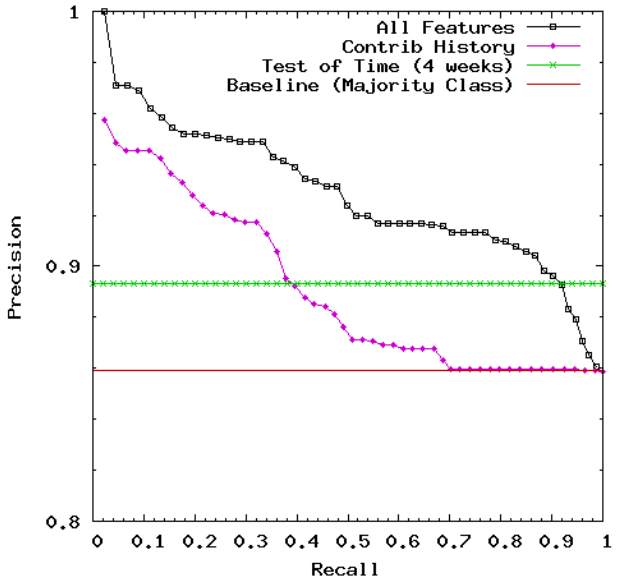


Figure 4: Precision vs. Recall for CQUAL, User Contribution History (Contrib History), the Majority Class baseline, and the "Test of Time" approach.

firming the informativeness of our feature sets. Logistic Regression consistently performs best, or nearly best, for all levels of recall; the performance is confirmed when considering the ROC curves. Therefore, we chose logistic regression for all our experiments and analyses.

4.3 Feature contribution analysis

Figure 6(a) reports the performance of classifiers using different feature groups individually. A classifier using user expertise features (Section 2.2.3) beats the baselines by a large margin. Features based on the contribution history of a user (Section 2.2.1) and those based on the prior difficulty of the different contribution types (Section 2.2.2) seem somewhat complementary, as the performance peaks and drops in different regions of the recall spectrum. Notably, the performance of the classifier that uses all three classes of features is significantly better, reaching 92.5% precision at 50% recall. Thus, for as many as half of the contributions we can make a highly accurate real-time decision whether the contribution is correct. The Precision-Recall results are complemented with analyzing the individual feature group performance through ROC curves (Figure 6(b)). Interestingly, Contribution History features only provide a lift over the majority baseline performance for low recall values (i.e., false positive ratio below 0.2), whereas Triple features (i.e., prior difficulty) accounts for a large lift for high recall levels (i.e., false positive ratio over 0.2). The ROC analysis demonstrates that these feature groups are somewhat complementary, and can be combined effectively into a single classifier (All Features). Interestingly, contribution history alone does not generate a significant improvement in performance compared to the naive "majority" baseline: The AUC value for the classifier that uses only contribution history features is at 0.543, barely above the random baseline of 0.5.

It is reasonable to anticipate that some of the feature groups are correlated, and individual feature group performance above may not contribute much to the final combined classifier. Therefore, we performed feature ablation analysis, by removing from the classifier one feature group at a time. Figure 7(a) contains the results. Interestingly, removing Triple Features (All-Triple Features in the graph) hurts performance the most, verifying that these features provide a unique signal not captured by contributor history or expertise. Somewhat puzzling is that removing Contributor History features actually *improves* performance slightly for some of the Recall levels. We conjecture that this is due to indiscriminately trusting a user who made correct contributions previously on one topic/domain to continue to make correct contributions in other domains, whereas modeling contributor *expertise* directly, as we do, does not suffer from this problem. The ROC curve results in Figure 7(b) reinforce this observation, indicating that occasionally ignoring contributor history (All-Contrib History in the graph) could improve performance. As expected, removing Triple and Contributor Expertise features consistently degrades performance. While some of the feature sets have very similar curves, based on our analysis of AUC, having all feature groups perform the best.

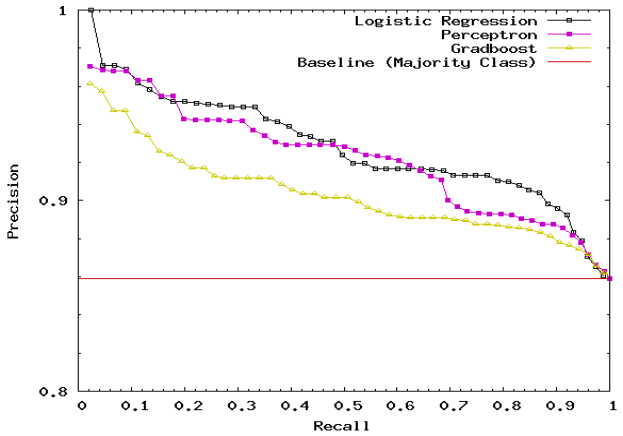
4.4 Analysis of Prediction Results

We now examine prediction results of each feature group to understand their performance in different scenarios, in order to provide more intuition about the contributions of different feature groups.

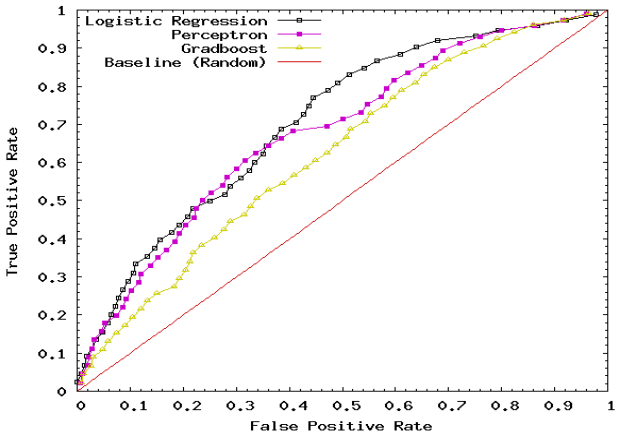
For example, the incorrect SPO triple[9] of <`/m/0kmyxvt, /film/film/initial_release_date, "2013-11-15"`>, was contributed by a user who exhibited historical accuracy of 0.88 prior to this contribution. If we had used merely the contribution history as our feature, our classifier would assign the confidence value of 0.8909 for this triple being correct, corresponding to the 55th percentile (i.e., 55% of the triples have lower classifier confidence), and would be predicted as "good". However, the other two feature groups perform better in this case, as they take into account of other signals orthogonal to the user's contribution history alone. For example, a classifier trained on triple features produced a low classifier confidence corresponding to 12th percentile (and thus would be labeled correctly as "bad"). This is likely because historically this predicate was observed to be generally challenging for "casual" Freebase users not intimately familiar with the domain. Similarly, a classifier trained using contribution expertise performs well, producing a relatively low classifier confidence of this triple, corresponding to 14th percentile (and thus would also be predicted as "bad"). This example illustrates a frequent scenario where topic- or predicate-specific features allow our classifier to outperform the contribution history-only baseline, by considering the domains and topics in which the user has previously demonstrated expertise.

Interestingly, contribution history does perform better than the other two feature groups in come cases. For example, for a correct SPO triple representing one of the Sea Otters in the Monterey Bay Acquarium, <`/m/0m55h, /zoos/zoo/`

[9]The triple is incorrect because the predicate /film/film/initial_release_date corresponds to the earliest release date of the film in any country. While the release date is 2013-11-15 in USA, this film will be released earlier in Russia on 2013-11-14.

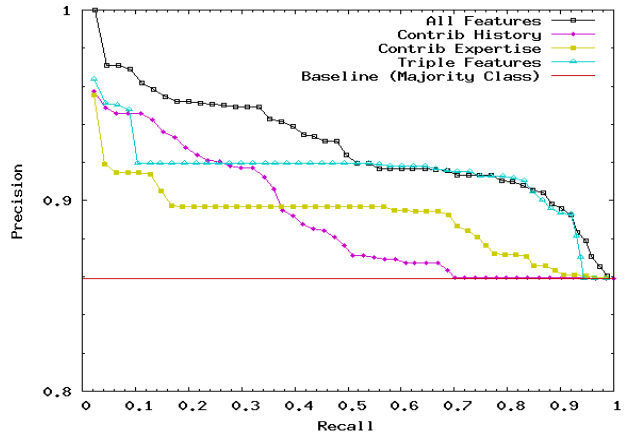


(a) Precision vs. Recall

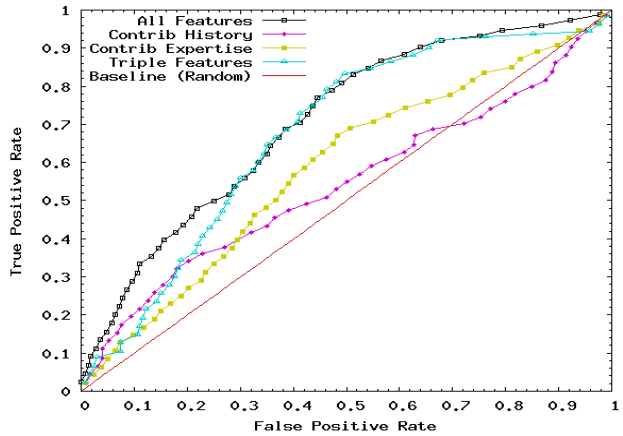


(b) ROC curves

Figure 5: Precision-recall and ROC curves for different learning algorithms.

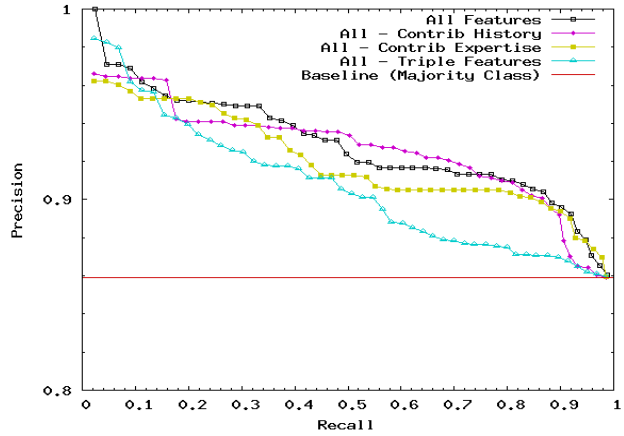


(a) Precision vs. Recall

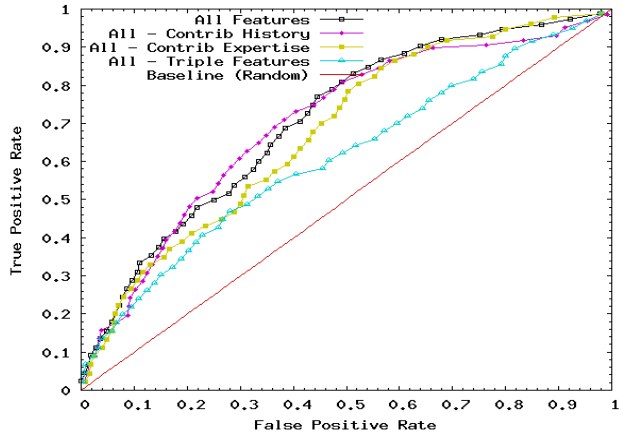


(b) ROC curves

Figure 6: Precision-recall and ROC curves for individual feature sets.



(a) Precision vs. Recall



(b) ROC curves

Figure 7: Precision-recall and ROC curves, removing one feature set at a time.

notable_animals, /m/0w4pq71>, the classifier trained using contribution history predicted a much higher confidence score compared to the classifier trained using triple features or contribution expertise alone – presumably as expertise in Sea Otters is not common among Freebase contributors. As different feature groups perform well in different scenarios, it is not surprising that our classifier, trained holistically by combining the signals from all three feature groups, consistently exhibits superior performance.

5. RELATED WORK

Quality Control in Crowdsourcing: In crowdsourcing settings, the typical way of inferring the quality of the contributors is either by using questions for which we already know the answer ("gold" data) or through redundancy, or by combining the two (see, for example, [31, 30, 35, 25, 19, 5, 16]). The main disadvantage of gold data is that it requires the user to go through a phase of contributing answers for topics that we already know, effectively wasting resources and preventing contributors from making original contributions. With redundancy, this problem is avoided, but it is still not possible to infer quickly the quality of the contributions, without waiting for multiple workers to make a contribution for the same topic. Our work sidesteps these problems by using certain features derived based on prior belief on the difficulties of the fact being contributed.

An orthogonal direction that also attempts to predict the quality of the work in real-time is the work by Rzeszotarski and Kittur [28] that use micro-behavioral signals, such as mouse movements, to predict the quality of the user submission. Although we do not record such level of user behavior, it is conceivable that such approaches could be seamlessly combined with our approach to further enhance the predictive power of our system.

Reputation systems: Reputation mechanisms [9, 10] are also commonly used as predictors of contribution quality. The most common form of reputation is to examine the behavior of the user in similar tasks in the past and try to predict future performance (e.g., [2, 27, 22]). In our work, we use such an approach as a baseline, and show that the incorporation of a wider set of features about the user can improve significantly the predictive accuracy of our approach.

Community question answering: A lot of research has been conducted in the past that aims to predict the quality of online answers in "Community Question Answering" (CQA) platforms [20, 24, 3, 1, 6, 32, 29]. The general setting there is to predict the quality of a submitted answer. Our work has three major conceptual and methodological differences: First, we use as predictors a set of signals that captures domain expertise of the user based on its prior contributions; this allows us to have a much better understanding of the knowledge that a user has about a topic, and indeed the addition of these signals improve significantly the predictive performance of our system. Second, we exploit the fact that Freebase contributions happen within a schema, and we generate estimates of difficulty of filling correctly a particular triple; we are not aware of any wok that tries to estimate the inherent difficulty of providing an answer to a question in a CQA site. Third, we are predicting the quality of *short, factual, and structured* contributions: Getting quality signals from such contributions is inherently different and more difficult than extracting quality signals from longer, textual pieces of text, where a variety of other features (e.g., readability, spelling, grammar) are providing useful signals for predicting the quality of the contribution.

Knowledge base construction: There is also work that describes the challenges incorporating and evaluating the human contributions in a knowledge base [2, 17, 36, 11, 27, 21]. For sites like Wikipedia, a challenge is to even measure the quality of contributions; longevity of the contributions is typically a good proxy for a high-quality contribution. Freebase allows a similar model to judge correctness, (contributions that survive for longer than a certain time threshold are considered correct). However, in our case we actually procured third-party editors to verify the accuracy of the contributions and did not assume that contributions that survived for long are also de-facto correct; we are not aware of similar third-party verification of KB facts in other work. In [36], Wick et al. describe how they use human contributions as just a piece of evidence, as opposed to considering the human edit to be correct by default; our work dovetails nicely with such efforts, as we are actively providing estimates of the trust that we should place in any user contributions, therefore facilitating the adoption of such KB construction schemes.

6. DISCUSSION

We presented a technique for *real-time*, *automatic* evaluation of submissions in a structured knowledge repository. Past techniques either operated on a time-delay basis (facts are considered correct only after a certain time period has passed and the fact has not been changed), or auto-approved changes based on the prior history of the contributor. Our technique operates in real-time, and examines contribution history in a holistic fashion: we model both the inherent difficulty of making contributions for specific types of facts and we also model the expertise of the user in various topical parts of the space. This leads to a significant reduction in error rates, while at the same time improving the timeliness of the information that is represented in the knowledge base.

In the future, we plan on leveraging more sources of information for modeling the probability that a fact contribution is correct or not. One possible future works is to use hard constraints based on schema rules, for example, we can use the fact that a person should have a date of death after a date of birth, and that two persons born on different eras could not possibly be spouse and etc. We can also use signals from other information extraction systems to infer the likelihood that a particular contribution is correct based on the evidences from the web. Similarly, we can use Item Response Theory (IRT) [14] to jointly model in a more fine-grained mode the ability of users in different knowledge domains, potentially allowing for *actively soliciting* users to contribute facts to areas that they are contributing already and are experts on. Combining this with an appropriate design of badges [4], or other rewards, could potentially allow for a more focused and faster generation of high-quality knowledge repositories.

Acknowledgments

We would like to thank John Giannandrea, Curt Janssen, Brian Karlak, and Kevin Murphy for helpful discussions and suggestions, and Jon Reitsma's team for providing the judgments.

References

[1] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge sharing and yahoo answers: Everyone knows something. In *WWW*, 2008.

[2] B. T. Adler, L. de Alfaro, I. Pye, and V. Raman. Measuring author contributions to the wikipedia. In *4th Int'l Symposium on Wikis*. ACM, 2008.

[3] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *WSDM*, 2008.

[4] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Steering user behavior with badges. In *WWW*, pages 95–106, 2013.

[5] Y. Bachrach, T. Graepel, T. Minka, and J. Guiver. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *arXiv preprint arXiv:1206.6386*, 2012.

[6] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *WWW*, 2009.

[7] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics*, 7(3):154–165, Sept. 2009.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[9] C. Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 2003.

[10] C. Dellarocas. Reputation mechanisms. In *Handbook on Economics and Information Systems*. Elsevier Publishing, 2006.

[11] O. Deshpande, D. S. Lamba, M. Tourn, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan, and A. Doan. Building, maintaining, and using knowledge bases: A report from the trenches. In *SIGMOD*, 2013.

[12] J. Duchi and Y. Singer. Boosting with structural sparsity. In *ICML*, pages 297–304, 2009.

[13] S. Dumais and H. Chen. Hierarchical classification of web content. In *SIGIR'00*, pages 256–263, 2000.

[14] S. E. Embretson and S. P. Reise. *Item response theory*. Psychology Press, 2000.

[15] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999.

[16] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *WSDM*, 2010.

[17] A. Halfaker, A. Kittur, R. Kraut, and J. Riedl. A jury of your peers: quality, experience and ownership in wikipedia. In *5th Int'l Symposium on Wikis and Open Collaboration*, 2009.

[18] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, Jan. 2013.

[19] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *KDD Workshop on Human computation*, pages 64–67, 2010.

[20] J. Jeon, W. B. Croft, J. H. Lee, and S. Park. A framework to predict the quality of answers with non-textual features. In *SIGIR*, 2006.

[21] S. Kochhar, S. Mazzocchi, and P. Paritosh. The anatomy of a large-scale human computation engine. In *KDD Workshop on Human Computation*, pages 10–17, 2010.

[22] M. Kokkodis and P. G. Ipeirotis. Have you done anything like that?: predicting performance using inter-category reputation. In *WSDM*, pages 435–444, 2013.

[23] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *ICML*, pages 170–178, 1997.

[24] Y. Liu, J. Bian, and E. Agichtein. Predicting information seeker satisfaction in community question answering. In *SIGIR*, 2008.

[25] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *JMLR*, 99:1297–1322, 2010.

[26] M. E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5:87–118, 2002.

[27] J. Rzeszotarski and A. Kittur. Learning from history: predicting reverted work at the word level in wikipedia. In *Computer Supported Cooperative Work*, pages 437–440, 2012.

[28] J. M. Rzeszotarski and A. Kittur. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Annual symposium on User interface software and technology*, pages 13–22. ACM, 2011.

[29] C. Shah and J. Pomerantz. Evaluating and predicting answer quality in community QA. In *SIGIR*, 2010.

[30] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, pages 614–622. ACM, 2008.

[31] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*, pages 254–263. Association for Computational Linguistics, 2008.

[32] M. A. Suryanto, E.-P. Lim, and A. S. R. H. L. Chiang. Quality-aware collaborative question answering: Methods and evaluation. In *WSDM*, 2009.

[33] F. B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *CHI*, pages 575–582, 2004.

[34] E. M. Voorhees. Overview of trec 2003. In *TREC*, pages 1–13, 2003.

[35] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. *NIPS*, 23:2424–2432, 2010.

[36] M. Wick, K. Schultz, and A. McCallum. Human-machine cooperation with epistemological dbs: supporting user corrections to knowledge bases. In *AKBC Workshop*, pages 89–94. ACL, 2012.