STEP: A Scalable Testing and Evaluation Platform

Maria Christoforaki
New York University
New York, NY
mc3563@nyu.edu

Panagiotis G. Ipeirotis
New York University
New York, NY
panos@stern.nyu.edu

Abstract

The emergence of online crowdsourcing sites, online work platforms, and even Massive Open Online Courses (MOOCs), has created an increasing need for reliably evaluating the skills of the participating users in a scalable way. Many platforms already allow users to take online tests and verify their skills, but the existing approaches face many problems. First of all, cheating is very common in online testing without supervision, as the test questions often "leak" and become easily available online together with the answers. Second, technical skills, such as programming, require the tests to be frequently updated in order to reflect the current state-of-the-art. Third, there is very limited evaluation of the tests themselves, and how effectively they measure the skill that the users are tested for.

In this paper, we present a Scalable Testing and Evaluation Platform (STEP), that allows continuous generation and evaluation of test questions. STEP leverages already available content, on Question Answering sites such as Stack Overflow and re-purposes these questions to generate tests. The system utilizes a crowdsourcing component for the editing of the questions, while it uses automated techniques for identifying promising QA threads that can be successfully re-purposed for testing. This continuous question generation decreases the impact of cheating and also creates questions that are closer to the real problems that the skill holder is expected to solve in real life. STEP also leverages the use of Item Response Theory to evaluate the quality of the questions. We also use external signals about the quality of the workers. These identify the questions that have the strongest predictive ability in distinguishing workers that have the potential to succeed in the online job marketplaces. Existing approaches contrast in using only internal consistency metrics to evaluate the questions. Finally, our system employs an automatic "leakage detector" that queries the Internet to identify leaked versions of our questions. We then mark these questions as "practice only," effectively removing them from the pool of questions used for evaluation. Our experimental evaluation shows that our system generates questions of comparable or higher quality compared to existing tests, with a cost of approximately $3-5$ dollars per question, which is lower than the cost of licensing questions from existing test banks.

Introduction

Today, increasingly skilled labor activities are carried out online. By connecting workers and employers through computer-mediated marketplaces, online labor markets such as Amazon Mechanical Turk, oDesk, and Mobileworks, can eliminate geographical restrictions, help participants find desirable jobs, guide workers through complex goals, and better understand workers' abilities. Broadly, online labor markets offer participants the opportunity to chart their own careers, pursue work that they find valuable, and do all this at a scale that few companies can presently deliver. Spurred by this revolution, some predict that remote work will be the norm rather than the exception (Davies, Fidler, and Gorbis 2011) within the next decade. One major challenge in this setting is to build skill assessment systems that can evaluate and certify the skills of workers, in order to facilitate the job matching process. Online labor markets currently rely on two forms of assessment mechanisms: *reputation systems* and *testing*.

Online markets often rely on reputation systems for instilling trust in the participants (Resnick et al. 2000; Dellarocas 2003). However, existing reputation systems are better-suited for markets where participants engage in a large number of transactions (e.g., selling electronics, where a merchant may sell tens or hundred of items in a short period of time). Online labor inherently suffers from data sparseness: many work engagements require at least a few hours of work, and many last for weeks or months. As a result, many participants have only minimal instances of feedback ratings which is a very weak reputation signal. Unfortunately, the lack of reputation signals creates a cold-start problem (Pallais 2013): workers cannot get jobs because they do not have feedback, and therefore cannot get feedback that would help them to get a job. In a worst case scenario, such markets may become "markets for lemons," (Akerlof 1970) forcing the departure of high-quality participants, leaving only low-quality workers as potential entrants. In offline labor markets, educational credentials are often used to signal the quality of the participants and avoid the cold-start problem (Spence 1973). In global online markets, credentialing is much trickier: verifying educational background is difficult, and knowledge of the quality of the educational institutions on a global scale is limited.

Given the shortcomings of reputation systems, many on-

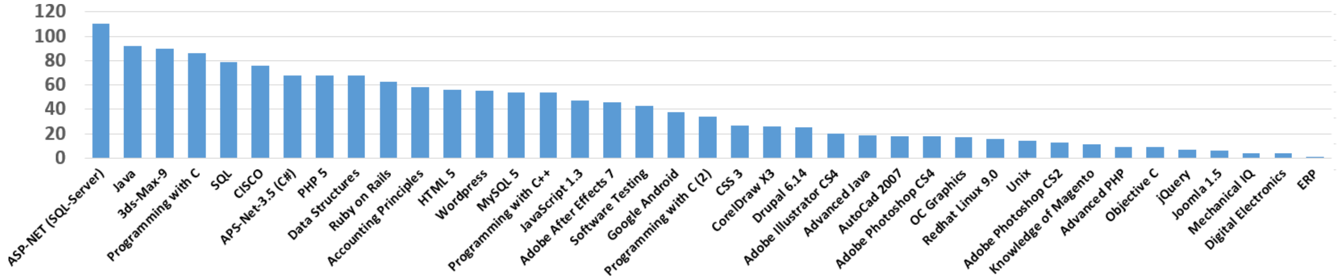


Figure 1: Number of URLs containing solutions to tests offered by oDesk, eLance, and Freelancer (the three biggest online labor marketplaces).

line labor markets resort to using testing as means of assessment, offering their own certification mechanisms. The goal of these tests is to verify/certify that a given worker indeed possesses a particular skill. For example, oDesk, eLance, and Freelancer allow workers to take online tests that assess the competency of these contractors across various skills (e.g., Java, CSS, Accounting, etc.) and then allow the contractors to display their achieved scores and ranking in their profile. Similarly, crowdsourcing companies such as CrowdFlower and Mechanical Turk certify the ability of contractors to perform certain tasks (e.g., photo moderation, content writing, translation) and allow employers to restrict recruiting to the population of certified workers. Unfortunately, online certification of skills is still problematic, with cheating being an ongoing challenge: since tests are available online, they are often “leaked” by some test takers and the answers become widely available on the web. Figure 1 illustrates a number of websites that contain solutions for the some of the popular tests[1] available on oDesk, eLance, and vWorker that we managed to identify, using simple web searches. Needless to say, the reliability of the tests for which answers are easily available through a web search is questionable.

Furthermore, it is common even for expert organizations to create questions with errors or ambiguities, especially if the test questions have not been properly assessed and calibrated with relatively large samples of test takers (Wingersky and Cook 1987). At the same time, many people question the value of the existing tests as long-term predictors of performance (Popham 1999; Jensen 1980; Newmann, Bryk, and Nagaoka 2001; Geiser and Santelices 2007; Fleming and Garcia 1998). The indications are that questions are calibrated only for *internal consistency* (how predictive a question is about the the final test score) and not for *external validity* (how predictive the question is for the long-term performance of the test taker). This question is particularly acute for online labor markets, as there is little research that examines whether testing and certifications are predictive of success in the labor market. Finally, as these test questions are presently created by independent experts, the quality of these questions relies heavily on the ability and inspiration of individuals, as opposed to having a systematic, reliable, and repeatable process, that can be used across organizations.

Crowdsourcing research has recently focused on techniques for getting crowd members to evaluate each other (Zhu et al. 2014; Dow et al. 2012). The hope is that peer assessment can lead to better learning outcomes as well (Kulkarni et al. 2013). Unfortunately, these systems still have large variance in final assessment scores, making them a poor match for certification and qualification.

In this paper, we alleviate some of the concerns of online testing by creating a system that: (a) is more cheat-proof than existing tests; (b) uses test questions that are closer to the real problems that a skill holder is expected to solve; and (c) assesses the quality of the tests using real market-performance data.

Our *Scalable Testing and Evaluation Platform (STEP)* leverages content generated on popular Q/A sites, such as StackOverflow, and uses these questions and answers as a basis for creating test questions. The use of real-life questions, allows the test questions generated to be (a) relevant to a real-world problem, and (b) continuously refreshed to replace questions that are leaked or outdated. Our system algorithmically identifies threads that are promising for generating high quality assessment questions, then uses a crowdsourcing system to edit these threads and transform them into multiple choice test questions. To assess the quality of the generated questions, we employ Item Response Theory and examine not only how predictive each question is regarding the internal consistency of the test (Embretson and Reise 2000) but also examines the correlation with future real-world market-performance metrics such as hiring rates, achieved wages, and others, using the oDesk marketplace as our experimental testbed for evaluation.

System Overview

Our STEP system consists of multiple components, as shown in Figure 2. Some components depend on human input[2] whereas others operate automatically. The life of a question in our system starts when extracting a promising Q/A thread from a Q/A-site. The thread is mapped to a particular skill and evaluated with respect to its appropriateness to serve as a test question. Thereafter it is edited, reviewed, and forwarded to the pool of testing questions.

[1] Sites such as `http://1faq.com/` and `http://www.livejar.info/`, are a couple of examples of the offenders.

[2] These components were assigned to oDesk contractors

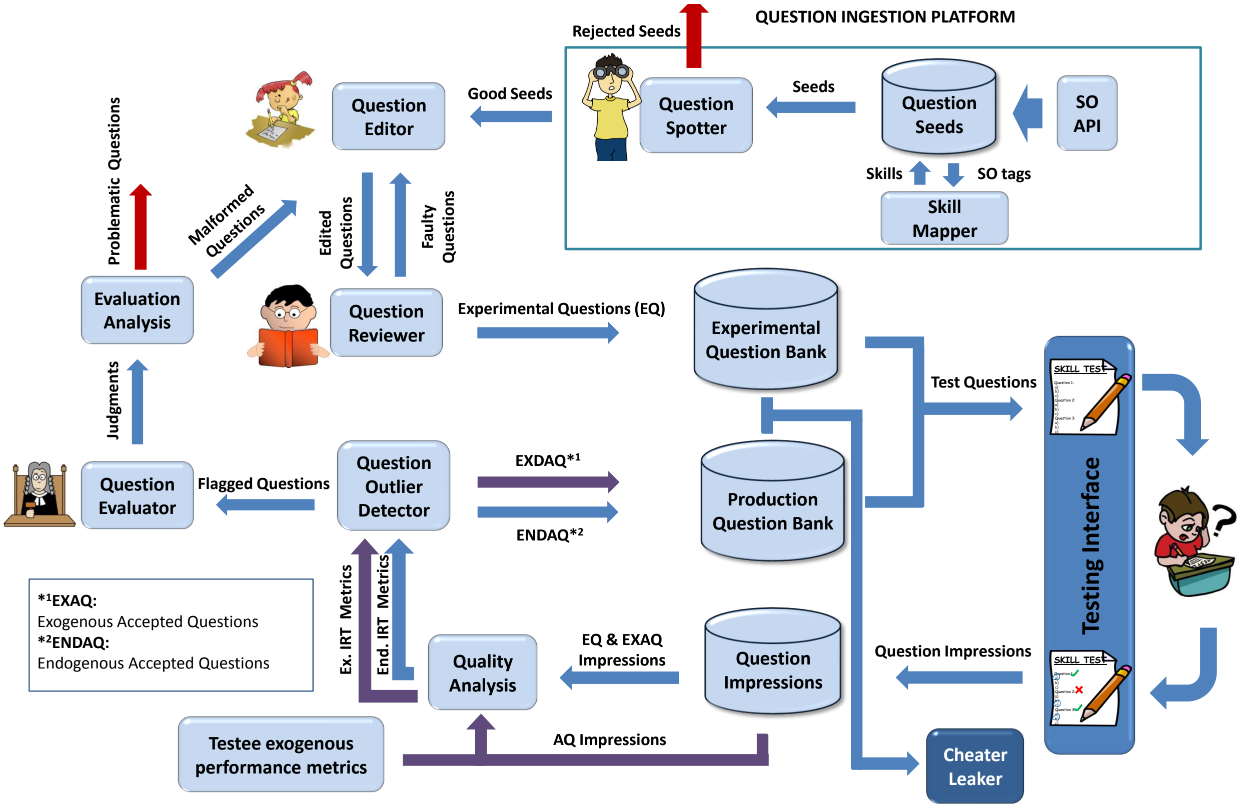


Figure 2: STEP Architecture and Components.

There, the question collects answer-responses from multiple users which are then used for its evaluation using Item Response Theory metrics. Depending on the outcome of the evaluation, the question is rejected, re-evaluated, or accepted. The accepted question metrics are used to accurately evaluate users with respect to their expertise in a particular skill.

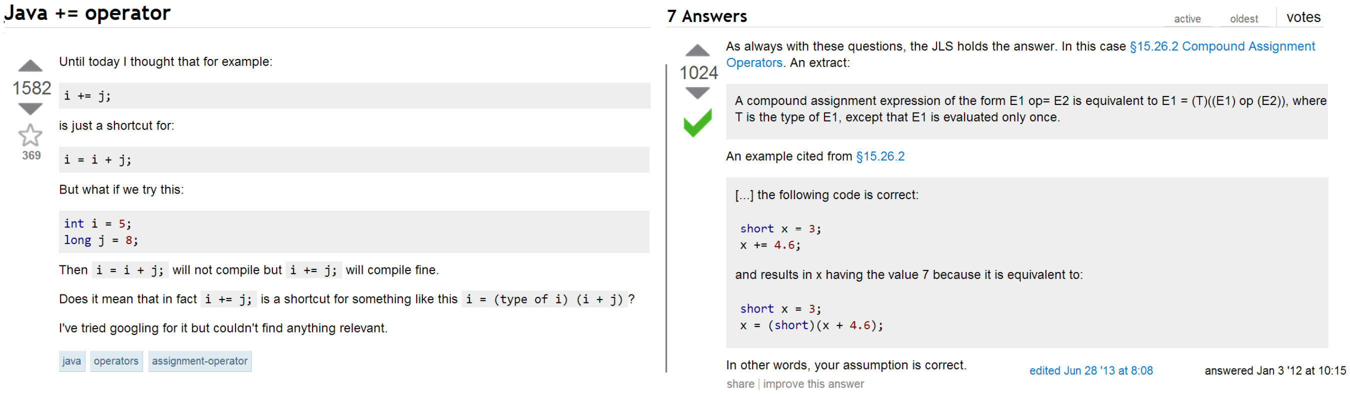
Question Ingestion Component: The Question Ingestion Component of STEP is responsible for collecting new "question seeds" from online resources in order to keep the question pool wide-ranging and fresh. In particular, the Ingestion component communicates with the Q/A site and fetches question and answer threads that are then stored in a database, together with a variety of metadata. The threads are labeled then as "promising" or not by an automatic classification model (see next section for details). The threads rejected by the classifier are removed from the question seed bank, whereas the accepted ones are forwarded to the editors to be transformed to standardized questions.

Reviewers are responsible only for checking spelling, syntactical errors, and compliance with the test formatting standards. Reviewers do not need to be domain experts but need to have a strong command of English to ensure that the questions have no spelling, syntactic, or grammatical errors. The editors and reviewers are hired, long-term contractors paid by the hour; they are not microtask-oriented workers, as in Mechanical Turk. As we point out in the "Experimental Evaluation" subsection, the percentage of STEP-generated test questions that gets accepted is higher than that for test questions acquired from test-generation companies. This provides a strong positive signal regarding the quality of the questions contributed by the Question Editors.

Question Editor and Reviewer: Q/A threads labeled as promising by the *Question Ingestion Component* are forwarded to the *Question Editors*. Question Editors are human contractors that are hired through oDesk. They are responsible for reformulating Q/A threads to match the style of a test question and adapt the answers to become choices of a multiple choice question.[3] The editors need to be domain experts, but not necessarily experienced in writing test questions (Q/A threads help in that respect).

Once the question is generated, then a *Question Reviewer* looks at the question. Reviewers are responsible only for checking for errors in writing and compliance with the test formatting standards (question text length, answer option count, answer text length, vocabulary usage etc.). They do not need to be domain experts but need to have strong command of English to ensure that the questions have no spelling, syntactic, or grammatical errors. The editors and reviewers are hired as long-term contractors paid by the hour; they are not microtask-oriented workers, as in Mechanical Turk. As we point out in the Experimental Evalua-

[3] For Q/A threads containing multiple valid answers, the question is often reformulated to "pick the best answer."



Java += operator

Until today I thought that for example:

```
i += j;
```

is just a shortcut for:

```
i = i + j;
```

But what if we try this:

```
int i = 5;
long j = 8;
```

Then `i = i + j;` will not compile but `i += j;` will compile fine.

Does it mean that in fact `i += j;` is a shortcut for something like this `i = (type of i) (i + j)`?

I've tried googling for it but couldn't find anything relevant.

7 Answers

As always with these questions, the JLS holds the answer. In this case §15.26.2 Compound Assignment Operators. An extract:

> A compound assignment expression of the form E1 op= E2 is equivalent to E1 = (T)((E1) op (E2)), where T is the type of E1, except that E1 is evaluated only once.

An example cited from §15.26.2

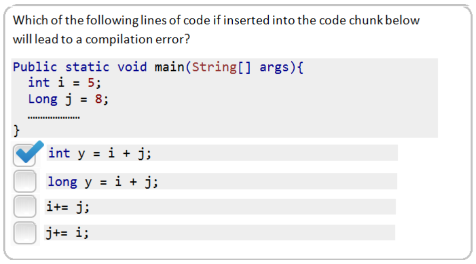
> [...] the following code is correct:
>
> ```
> short x = 3;
> x += 4.6;
> ```
>
> and results in x having the value 7 because it is equivalent to:
>
> ```
> short x = 3;
> x = (short)(x + 4.6);
> ```

In other words, your assumption is correct.

edited Jun 28 '13 at 8:08

answered Jan 3 '12 at 10:15

Figure 3: Example of Q/A Thread (left) transformation to a multiple choice Java Test Question (right).



Which of the following lines of code if inserted into the code chunk below will lead to a compilation error?

```
Public static void main(String[] args){
  int i = 5;
  Long j = 8;
  ....................
}
```

☑ `int y = i + j;`

☐ `long y = i + j;`

☐ `i+= j;`

☐ `j+= i;`

Figure 4: Test Question created from Figure 3 Q/A thread.

tion subsection, the percentage of STEP-generated test questions that gets accepted is in fact higher than that for test questions acquired from test-generation companies, which provides a strong positive signal regarding the quality of the questions contributed by the Question Editors.

Each question approved by the reviewer becomes *Experimental* and is committed to the Experimental Question Bank. Non-approved questions are sent back to the Question Editor for re-editing. Figure 3 shows an example of a Stack Overflow Java Q/A thread and Figure 4 illustrates its transformation of into a test question.

Question Bank: Experimental and Production The Experimental Question Bank stores questions that are created by the question editor, but are not yet evaluated. The experimental questions are included in the tests but are only 10% to 20% of the questions, and are not being used for the evaluation of the users. Once the experimental questions receive enough answers, they are forwarded for evaluation to the *Quality Analysis* component. The Production Question Bank stores those questions that are shown to users in tests and that are used for their evaluation. Production Questions are also evaluated periodically using the Quality Analysis component.

Quality Analysis: The Quality Analysis Component is the part of the system that is responsible for computing quality metrics for each question. Its main functionality is the quality evaluation of the test questions. The experimental questions are evaluated using "endogenous" metrics (i.e., whether the performance of the users in that question correlates well the overall test score), and if they perform well graduate into production. The production questions are evaluated periodically using exogenous metrics (i.e., how well they can predict the market performance of the users a few months after the test). We describe the process in detail in the corresponding section below.

In addition to calculating the quality metrics, the component also has an outlier detector that identifies questions that behave differently than others; such questions are forwarded to human experts that examine whether the question has any technical error, ambiguity, and so on. Problematic questions that can be corrected are edited and reintroduced in the system as experimental questions. Ambiguous and irrelevant questions are typically discarded, as they are difficult to fix. A question is also discarded if no particular problem has been identified but the question still exhibits unusual behavior. A common cause for the problematic behavior is that the question has been compromised. Even if the question is correctly formulated, and theoretically is able to discriminate test takers with different ability levels, when it has leaked, a user's answer to this question is not a reliable signal for the user's ability in the topic, leading to strange statistical behavior.

Cheater Leaker: The *Cheater Leaker* component queries continuously queries against popular search engines, monitoring for leaked versions of the test questions. [4] Once a question is located "in the wild," a Question Editor visits the identified web site and examines whether indeed it contains the question and the answers. A question is then marked as "leaked" and gets retired from the system: the leaked questions are released as practice questions and teaching/homework material for learning the skill. This component is also used to ensure that when the question is originally created by the editor, it is sufficiently reworded to avoid being located by simple web queries.

[4] The techniques we used for detecting highly similar documents on the web involve the use of "unusual" n-grams as queries in search engines, to detect pages with similar content. We use both existing commercial services for approximate querying (e.g., CopyScape) and "query by document" techniques (e.g., (Yang et al. 2009)).

The main goal of the Cheater Leaker is to prevent test-takers from searching the question or part of a question online and directly finding the correct answer option in certain forums. If the question was not reformulated to be significantly different from the original that was found in the Q/A thread, or is still similar to its older version that had been leaked, this is detected by the Cheater Leaker. People taking an online test face a time constraint of slightly more than one minute per question on average hence, a test-taker can take advantage of a leak only if a) she can find it quickly, b) she can directly interpret the answer she sees online into the appropriate answer in the test. As we will discuss in the Question Quality Evaluation Section, even if a question has been leaked and cannot be identified by the Cheater Leaker, but workers somehow are able locate and usethe leaked answers consistently, this will be identified in the long run by the Question Evaluation component since the discrimination will gradually decrease, especially for the exogenous metrics of ability.)

Question Generation Process

Our system leverages existing Question Answering sites, to generate seeds for new test questions. The volume of the available questions in sites such as StackOverflow is both a blessing and a curse: The large number of questions gives us many seeds for generating questions; however only a small fraction of the QA threads are suitable for the generation of test questions and we need to identify the most promising threads to avoid overwhelming the editors with false leads.

Stack Exchange

Stack Exchange is a network of more than a hundred sites with Question Answer threads on different areas ranging from software programming questions to Japanese Language and Photography questions. SE provides an API and provides programmatic access for downloading questions posted on these platforms along with all the answers and comments associated with them as well as a number of other semantically rich question, answer, and comment features, like view count, up votes, down votes, author reputation scores and so on. The downloaded questions are separated into topics by leveraging the tags attached to each question.

Our current system focuses on testing for technical skills and therefore we focus on Stack Overflow. Stack Overflow is Stack Exchange's most popular site and it is "a question and answer site for professional and enthusiast programmers". It has almost 3 million subscribed users and more than 6 million questions associated with 35K tags. Table 1 shows the 10 most popular topics which compose slightly more than 20% of the total volume of questions. Needless to say, it is not feasible or desirable to manually examine all threads to examine which threads are the most promising for generating test questions. Ultimately, we want to automate the process of identifying good threads and then use them as seeds for question generation. Ideally, the question should test something that is confusing to users when they learn a skill, but clear for experts.

Topic	Questions	Percentile (%)
C#	508,194	3.08
Java	468,554	2.84
PHP	433,801	2.63
Javascript	433,707	2.63
Android	377,031	2.29
Jquery	355,800	2.16
C++	222,599	1.35
Python	216,924	1.32
HTML	198,028	1.20
mysql	184,382	1.12

Table 1: Top-10 popular Stack Overflow tags

Question Spotter

Towards this goal, we follow a three-stepped approach for labeling threads as good or not. As a tradeoff between speed and reliability of labeling, *each thread is assigned three labels, that mark whether it is a good QA thread.* The three labels corresponds to tradeoffs between the timeliness of creating the label and the corresponding reliability of the label that indicates whether the thread is a good one for test question generation. The first, label comes through crowd voting, where five workers look at the QA thread and vote on whether the thread is promising for generating a test question; this label is rather noisy but quickly helps us remove non-promising threads from consideration. The other two labels are generated by the Quality Analysis Component, and correspond to whether the question that was generated by the thread ended up being of high quality and whether it had predictive value in predicting the future performance of the test-taker.

Using the three labels described above, we then build automatic classification models that assign a label to each incoming QA thread. We endow each QA thread with a set of features, such as number of views, number of votes for the question and each of the answers, the entropy of the vote distribution among the answers, the number of references to the thread, the tags assigned to the text, the length of the question text and of the answers, the number of comments, the reputation of the members that asked the question and gave the answers, and so on.

We built the classifier using Random Forests, and our objective was to optimize for the precision of the results and minimize the number of false positives in the results (i.e., minimize the bad threads listed as good). Our achieved precision ranged from 90% to 98% across a variety of technical topics. This measurement is based on how many of the presented seeds were selected by the question editors and transformed into questions.

We also performed a qualitative assessment of the features used to get a better understanding of what makes a QA thread a good seed for a test question. We noticed that a large number of upvotes is actually a *negative* predictor for suitability for the thread to generate good test questions: highly voted questions tend to ask about arcane topics with little practical value. On the other hand, threads with a large number of answers and high-entropy distribution of upvotes

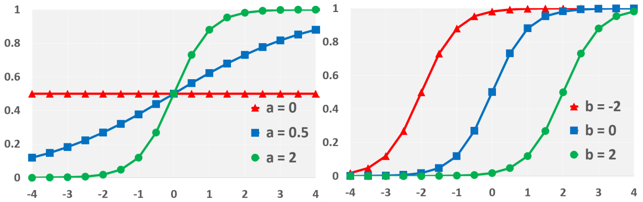


Figure 5: Illustrations of the 2PL "item characteristic curve" for different discrimination (left) and difficulty (right) values. X-axis is the normalized ability θ and Y-axis is the probability $P(\theta)$ that a person with that ability will answer the question correctly.

across the answers, signal the existence of a topic that is confusing users, with many answers that can serve as "distractor answers" (Guttman and Schlesinger 1967). We also found found that question threads frequently visited by many users indicate questions on common problems for a variety of expertise levels for the topic at hand. Also the number of incoming links to the question are highly correlated with high-quality answers, while threads with very long answers are also not good for test-question generation, even if they get large number of upvotes. Of course, the true question is not the predictive ability of the question spotter component, but rather how many of the questions inspired by the seeds ended up being good test questions. We discuss that topic in the next section.

Question Quality Evaluation

The Question Analysis component of our system generates a set of metrics to evaluate the quality of the questions in the Question Banks. We compute these metrics using standard methods from *Item Response Theory (IRT)*, a field of psychometrics for evaluating the quality of tests and surveys to measure abilities, attitudes, and so on. The prerequisite for analyzing a question (the "Item" in IRT) is for the question to be answered by a sufficiently large number of test-takers. Once we have that data, IRT can then be used to examine how well the test question measures the "ability" θ of a test-taker. Traditionally, the θ is approximated by the score of the user in the overall test, and is rather "endogenous." As a key contribution of our system, in addition to the endogenous measure of ability, we also use "exogenous" market performance metrics for measuring the ability θ of a test-taker as demonstrated in the market, and not just based on the test results.

Basics of Item Response Theory

Before describing our question evaluation process in detail, we briefly discuss some preliminaries on Item Response Theory (Ronald K. Hambleton 1991). The first assumption in IRT is that the test-takers have a single ability parameter θ, which represents the subject's ability level in a particular field, which customarily we consider to have a $N(0, 1)$ normal distribution, with the population mean having $\theta = 0$. The second assumption is that items are conditionally independent, given an individual's ability. Given these two assumptions, the basic concept of IRT is that each question can be characterized by the probability $P(\theta)$ that a user with an ability θ will give a successful answer to the question. This function $P(\theta)$ is called *Item Characteristic Curve* (ICC) or *Item Response Function* (IRF) and has the following general form:

$$P(\theta) = c + \frac{d - c}{1 + e^{-a(\theta - b)}} \qquad (1)$$

The parameter a is called *discrimination* and quantifies how well the question discriminates between test-takers with different ability levels. Higher values of a result in a steeper curve, which means that the probability of answering correctly increases sharply with the ability of the test taker. The parameter b is called *difficulty*; it corresponds to the value of θ where $P(\theta) = 0.5$ and is also the inflection point of the curve. Higher values mean that only high ability test-takers answer the question correctly. Finally, c is the probability of guessing the correct answer randomly for each question and d is the highest possible probability of answering a question correctly. For simplicity, customarily we set $c = 0$ for free-text answers or $c = 1/n$ for multiple choice questions, with n being the number of available answers, and we set $d = 1$.

Figure 5 illustrates how the ICC changes for different values of discrimination and difficulty. On the left, the question's difficulty is set to zero and the lines show the ICC for three discrimination values. When the discrimination is zero, the line is flat and it is obvious that there is no correlation between the test-taker ability and the probability of answering the question correctly. On the right plot, the question's discrimination is set to 2 and the three lines show the ICC for three difficulty values. Smaller difficulty values shift the steep part of the curve to the left and let test takers with lower ability levels have better chances of answering the question correctly.

An important additional metric to consider is the *Fisher information* $I(\theta)$ of the $P(\theta)$ distribution. From the Wikipedia description "Fisher information is a way of measuring the amount of information that an observable random variable X carries about an unknown parameter θ upon which the probability of X depends." In our context, the Fisher information of a question shows how accurately we can measure the ability θ (the unknown parameter) for a user after observing the answer to the question (the observed random variable). Formally:

$$I(\theta) = a^2 \frac{e^{-a(\theta - b)}}{(1 + e^{-a(\theta - b)})^2} \qquad (2)$$

In general, highly discriminating items have tall, narrow information functions and they can measure with accuracy the θ value but over a narrow range. Less discriminating questions provide less information but over a wider range. Intuitively, highly discriminative questions can provide a lot of information about the ability of a user around the inflection point (as they separate the test takers well) but are not providing much information in the flatter regions of the curve.

An important and useful property of Fisher information is its additivity. The Fisher information of a test is the sum

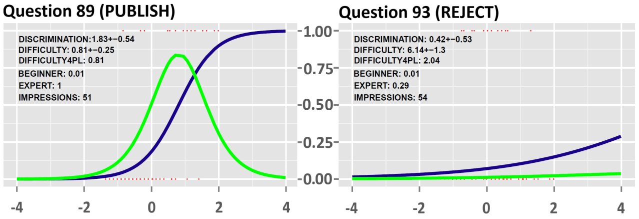


Figure 6: Example of an accepted (left) vs. a rejected experimental question. X-axis is the test taker ability θ and Y-axis $P(\theta)$ (blue) and $I(\theta)$ (green).

of the information of all the questions in the test. So, when creating a test, we can select questions with that have high $I(\theta)$ across a variety of θ values to be able to measure well the ability θ across a variety of values. Of course, if we want to measure more accurately some regions, we can add more questions that have high $I(\theta)$ for the regions of interest.[5]

Question Analysis based on Endogenous Metrics

Following the paradigm of traditional IRT, our first quality analysis uses as a measure of the ability θ of the test score of the test-taker, computed over only the production questions in the test (and not the experimental). The raw test score for each user i is then converted into a normalized value θ_i, so that the distribution of scores is a standard normal distribution.[6] Once we have the ability scores θ_i for each user i, we then analyze each question j. The answer of the user in each question is binary, either correct or incorrect. Using the data, we fit the ICC curve and we estimate the discrimination a_j and the difficulty b_j for each question.

For an experimental question to move to production, we require the discrimination to be in the top-90% percentile across all questions, and of course to be positive. Figure 6 shows the ICC and information curves for two questions. An accepted question has a high discrimination value, and correspondingly high Fisher information; a rejected question typically has low discrimination and low Fisher information. When analyzing existing tests, we also observed questions with high but *negative* discrimination values; these questions almost always had an incorrect answer marked as correct, or were "trick" questions testing very arcane parts of the language. Figures 7 and 8 show the ICC and information curves of two questions about Java. The blue curve illustrates the ICC curve and the green curve the information curve. Figure 7 shows a question with high discrimination and medium difficulty, whereas Figure 8 shows a question with high difficulty and low discrimination.

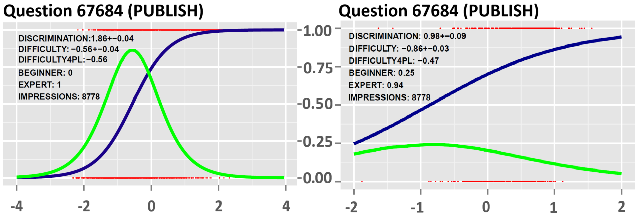


Figure 7: Example of accepted Production Question Analysis based on endogenous (left) vs. exogenous (right) metrics. The X-axis is the test-taker ability θ and the Y-axis is the probability $P(\theta)$ of answering the question correctly (blue curve) and $I(\theta)$ is the Fisher information (green curve).

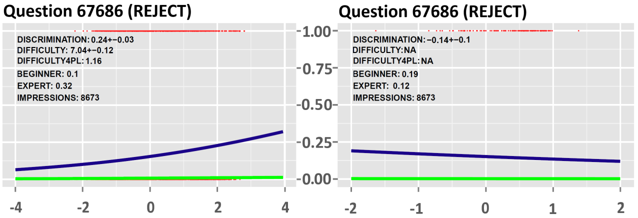


Figure 8: Example of rejected Production Question based on endogenous (left) vs. exogenous (right) metrics. The X-axis is the test-taker ability θ and the Y-axis is the probability $P(\theta)$ of answering the question correctly (blue curve) and $I(\theta)$ is the Fisher information (green curve).

Question Analysis based on Exogenous Metrics

A common complaint about tests is that they do not focus on topics that are important "in the real world." As an important contribution of our work, we decided to also use "exogenous" ability metrics to represent the test-taker θs. Exogenous ability metrics measure the success of the test-taker in the labor market, as opposed to the success while taking the test. Examples of these metrics are the test-taker's average wage, her hiring rate, the jobs that she has completed successfully etc. Using exogenous metrics makes the evaluation of the questions more robust in discovering cheating, and can indicate more easily which of the skills tested by the question are also important in the marketplace. For brevity, in this paper, we present the results using the log of wages 3 months after the test, to represent the test taker's ability θ.

Not surprisingly, the questions do not exhibit the same degree of correlation with the exogenous user abilities compared to the endogenous ability (the user test-score itself). The right plot in Figure 7 shows the ICC and information curves of the same question as the left plot but computed using the exogenous ability metrics. We observe that the discrimination of the question that was computed using the endogenous ability metrics provides relatively high discrimination (0.98) but still not as high as the the discrimination computed using the exogenous metrics (1.86). The same holds for the two plots in Figure 8. Both plots show a low

[5] Typically, we want to measure accurately the ability of the top performers while we are rather indifferent when separating the bottom-50%. Unfortunately, in reality, it is difficult to construct many test questions that have *both* high discrimination *and* high difficulty.

[6] Instead of allowing all questions to contribute equally to the raw score, some IRT algorithms allow each question to contribute differently to the score, according to the discrimination power and the difficulty. Although more principled, the changes in the scores are often negligible with more than 95% of the scores remaining the same and with the additional problem that it is not possible to explain the scoring mechanism to the students.

quality question, with the discrimination computed by the exogenous ability metrics actually being negative. The pattern holds across all questions that we have examined. One immediate, practical implication is that we need more test-takers to be able to robustly estimate the discrimination and difficulty parameters for each question.

Our analysis with an exogenous ability metric has two objectives. First, we better understand the contractors and their ability to perform well in the marketplace. Second, we also determine which of the test questions are still useful for contractor evaluation: for questions that are leaked, or questions that are now outdated (e.g., deprecated features), the exogenous evaluation shows a drop of discrimination over time, giving us signals that the question has to be removed or corrected.

Experimental evaluation

Our approach for generating tests from QA sites has the clear advantage of being able to generate new questions quickly, compared to the existing practice of using a "static" pool of test questions. However, there are two key questions when considering this approach: (a) How do the questions perform compared to existing test questions, and (b) What is the cost for generating these questions?

In order to evaluate the benefit of our system compared to the existing approach of using a static question bank, we generated test-questions with STEP for the following skills: PHP, Python, Ruby on Rails, CSS, HTML, and Java. Our test-takers are contractors registered with oDesk, who took the tests to certify their skills. oDesk already gives contractors the option of taking skill tests, which are generated by external test-generation companies; each of these tests comprises 40 questions (administered with a time limit of 60 minutes). We inserted 5 questions generated by STEP to each such test to evaluate the quality of the questions generated by STEP and collected at least 100 responses for each STEP-generated question. Clearly, the STEP-generated questions were not included for the oDesk user skill certification process. We also had access to the exogenous metrics of oDesk users to evaluate our methods. Our experiments were conducted with test-taker wages that were provided to us by oDesk. We also experimented with other metrics (e.g., hiring rate). The differences were not big and wages seemed to be the best and more robust exogenous metric of "success" in the oDesk platform, so we picked that one for our experiments.

Hence, for each skill we had the existing test that contained questions from a "static" question bank, generated by domain experts, and the new STEP test, which contained only questions generated by our STEP system, using StackOverflow threads.

For both of these two tests, we computed the information curve for the test, by summing the information gain of all its questions. The Fisher information gain is considered one of the standard metrics for measuring question quality, hence we focused on that as the metric of performance.

Figure 9 displays the results for the Java test; the results were very similar for all the other skills that we experimented with (PHP, Python, Ruby on Rails, CSS, HTML, Java). The left plot shows the information curve for the test containing the "static question bank" questions; the right the information gain for the test containing the STEP-generated questions. The x-axis is the ability level of the test takers and the y-axis the information of the test for the particular ability level. As a reminder, high information values mean higher precision of the test when measuring the ability of a worker with a certain ability. Both tests behave similarly, indicating that our STEP questions have the same quality on average as the questions that are generated by domain experts.

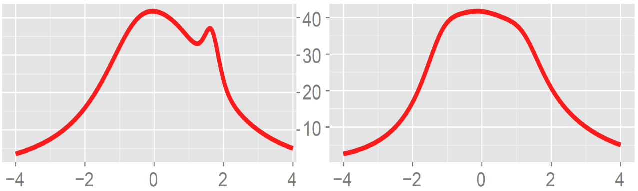


Figure 9: Information curves for a test containing questions generated by domain experts (left), vs. new a test with STEP-generated questions inspired by StackOverflow threads. X-axis is the test-taker ability θ and the Y-axis is the Fisher information $I(\theta)$.

We also examined how many of the questions in the two tests were able to pass the evaluation that used the exogenous ability (wage) as the ability metric. When evaluating the domain expert questions, 87% of the questions were accepted, whereas the STEP questions have a 89% acceptance rate. The numbers are roughly equivalent, indicating that STEP can generate questions at the same level of quality (or even higher) than the existing solutions.

Given that the quality of the STEP tests is equivalent to the existing tests that we can acquire from a question bank, the next question is whether it makes financial sense to create questions using STEP. The cost of the question in STEP ranged from \$3/question to \$5/question, depending on the skill tested, with an average cost of \$4/question. For the domain-expert questions, the cost per question was either a variable \$0.25/question *per user taking the test* or \$10 to buy the question[7]. Therefore, it is also financially preferable to use STEP to generate questions compared to using existing question banks; in addition to being cheaper, STEP also allows for a continuous refreshing of the question bank, and allows the retired questions to be used by current users as practice questions for improving their skills.

Discussion & Future Work

We presented STEP, a scalable testing and evaluation platform. Our system leverages content from user-generated Question Answering websites to continuously generate test questions, allowing the tests to be always "fresh", minimizing the problem of question leakage that unavoidably leads to cheating. We also show how to leverage Item Response Theory to perform quality control on the generated questions and, furthermore, we use marketplace-derived metrics

[7] The numbers correspond to \$10 per user taking a 40-question test, or \$500 to buy the full question bank that contained 50 questions.

to evaluate the ability of test questions to assess and predict the performance of contractors in the marketplace, making it even more difficult for cheating to have an actual effect in the results of the tests.

One important direction for the future is to build tests that have higher discrimination power for the top-ranked users than for the low-ranked ones (e.g., discriminate better between the top-5% and top-20%, compared to between the bottom-5% and bottom-20%). We expect the use of adaptive testing to be useful in that respect as we can have tests that terminate early for the low-ranked users, while for the top-ranked users, we may ask more questions, until reaching the desired level of measurement accuracy.

Furthermore, we want to apply STEP for generating tests for non-programming skills by leveraging non-technical QA sites, and even generate tests for MOOCs by analyzing the contents of the discussion boards, where students ask questions about the content of the course, the homework, etc. We believe that such a methodology will allow the tests to be more tailored to the student population and that can measure better the skills that are expected in the marketplace.

Platforms like Mechanical Turk could also benefit from a system like STEP. The typical tasks on such platforms may not require deep technical skills like the knowledge of a programming language, hence skill testing is less applicable in that context. However, the evaluation mechanism can be leveraged for typical MT tasks to identify "golden" questions that can discriminate well between "good" and "bad" workers for the task at hand.

Acknowledgements

The authors would like to thank Samir Lakhani and Thomas Eythimiou for their help, and also oDesk for providing access to their platform. Maria Christoforaki has been supported by NSF III-1117829, and Panos Ipeirotis has been supported by a Google Focused Award.

References

Akerlof, G. A. 1970. The market for "lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics* 488–500.

Davies, A.; Fidler, D.; and Gorbis, M. 2011. *Future work skills 2020.* Institute for the Future for University of Phoenix Research Institute.

Dellarocas, C. 2003. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science* 49:1407–1424.

Dow, S.; Kulkarni, A.; Klemmer, S.; and Hartmann, B. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, 1013–1022. New York, NY, USA: ACM.

Embretson, S. E., and Reise, S. P. 2000. *Item Response Theory*. Psychology Press.

Fleming, J., and Garcia, N. 1998. Arwwe standardized tests fair to african americans?: Predictive validity of the sat in black and white institutions. *Journal of Higher Education* 471–495.

Geiser, S., and Santelices, M. V. 2007. Validity of high-school grades in predicting student success beyond the freshman year: High-school record vs. standardized tests as indicators of four-year college outcomes. Technical report, University of California–Berkeley.

Guttman, L., and Schlesinger, I. 1967. Systematic construction of distractors for ability and achievement test items. *Educational and Psychological Measurement* 569–580.

Jensen, A. R. 1980. *Bias in Mental Testing.* ERIC.

Kulkarni, C.; Wei, K. P.; Le, H.; Chia, D.; Papadopoulos, K.; Cheng, J.; Koller, D.; and Klemmer, S. 2013. Peer and self assessment in massive online classes. *Computer-Human Interaction* (39):33:1–33:31.

Newmann, F. M.; Bryk, A. S.; and Nagaoka, J. K. 2001. *Authentic intellectual work and standardized tests: Conflict or coexistence?* Consortium on Chicago School Research Chicago.

Pallais, A. 2013. Inefficient hiring in entry-level labor markets. Technical report, National Bureau of Economic Research.

Popham, W. J. 1999. Why standardized tests don't measure educational quality. *Educational Leadership* 56:8–16.

Resnick, P.; Kuwabara, K.; Zeckhauser, R.; and Friedman, E. 2000. Reputation systems. *Communications of the ACM* 43(12):45–48.

Ronald K. Hambleton, Hariharan Swaminathan, H. J. R. 1991. *Fundamentals of Item Response Theory*. SAGE, 3 edition.

Spence, M. 1973. Job market signaling. *The Quarterly Journal of Economics* 87(3):355–374.

Wingersky, M. S., and Cook, L. L. 1987. *Specifying the characteristics of linking items used for item response theory item calibration*. Educational Testing Service.

Yang, Y.; Bansal, N.; Dakka, W.; Ipeirotis, P. G.; Koudas, N.; and Papadias, D. 2009. Query by document. In *WSDM*, 34–43.

Zhu, H.; Dow, S. P.; Kraut, R. E.; and Kittur, A. 2014. Reviewing versus doing: Learning and performance in crowd assessment. In *Proceedings of the ACM 2014 Conference on Computer Supported Cooperative Work*. ACM.