Capturing Winning Ideas in Online Design Communities

Faez Ahmed and Mark Fuge
University of Maryland
College Park, USA
{faez00,fuge}@umd.edu

ABSTRACT

This paper describes how to find or filter high-quality ideas submitted by members collaborating together in online communities. Typical means of organizing community submissions, such as aggregating community or crowd votes, suffer from the cold-start problem, the rich-get-richer problem, and the sparsity problem. To circumvent those, our approach learns a ranking model that combines 1) community feedback, 2) idea uniqueness, and 3) text features—e.g., readability, coherence, semantics, *etc.* This model can then rank order submissions by expected quality, supporting community members in finding content that can inspire them and improve collaboration among members.

As illustrative example, we demonstrate the model on OpenIDEO—a collaborative community where high-quality submissions are rewarded by winning design challenges. We find that the proposed ranking model finds winning ideas more effectively than existing ranking techniques (comment sorting), as measured using both Discounted Cumulative Gain and human perceptions of idea quality. We also identify the elements of winning ideas that were highly predictive of subsequent success: 1) engagement with community feedback, 2) submission length, and 3) a submission's uniqueness. Ultimately, our approach enables community members and managers to more effectively manage creative stimuli created by large collaborative communities.

Author Keywords

OpenIDEO; Online Communities; Crowdsourcing; Recommender Systems; Classification; Ranking

ACM Classification Keywords

H.5.3. Information Interfaces and Presentation: Group and Organization Interfaces; Computer-supported cooperative work

INTRODUCTION AND RELATED WORK

Online communities have changed the way people collaborate and work together: thousands of people can pursue goals together at a previously impossible scale. For example, in online communities, members can build off of high quality ideas submitted by others to produce new knowledge bases (*e.g.*, Wikipedia), answer questions [38], create music [36], and even produce real-world products and services—*e.g.*, OpenIDEO, Threadless, Local Motors, *etc.* [5, 22, 30].

This scale, however, creates new challenges for cooperative work. For example, in a sea of thousands, how can someone sift through ideas to find high quality submissions they can build upon or be inspired by? How does one find the needles in the submission haystack? (Or at least remove vast quantities of hay before one starts searching.) We review both human and automated techniques to do this, and then propose and validate a model that combines the strengths of both. Specifically, this paper focuses on applications to online design communities—communities where members collaborate together on designing a new product or service—because those communities rely on being able to build upon and become inspired by high quality ideas. However, the approach used here could also extend to other collaborative work communities where filtering submissions by a quality metric could improves work outcomes.

Leveraging Humans

One common approach to managing community submissions is to use those same communities on themselves: leverage crowds of people to both submit ideas as well as then filter and sort ideas. The simplest and most common approach asks the crowd to vote on ideas and then orders ideas by vote count. While easy to implement and widely used, it struggles in several cases: 1) The *cold-start problem*—if ideas have no votes yet, how does one initially sort ideas or assign them to members for voting? [23] 2) The *rich-get-richer problem*—if you cannot force members to see randomized submissions, members may only visit (and vote on) already highly voted ideas, thus biasing vote counts [21]; and 3) The *sparsity problem*—even if you have a large, involved community with randomized assignments, members may still never assess some ideas due to fatigue or lack of bandwidth, leaving a large number of ideas without any ratings.

Past work has sought to use humans to mitigate the above problems in two complementary ways. First, one might break down a complex task (like understanding, organizing, and rating ideas) into several smaller, simpler, and compartmentalized tasks that can be handled by additional workers—for example, assembling document summaries [3] or topical outlines [27]. Such approaches are effective provided such a task decomposition is possible, and that one can access and support a sufficiently large pool of qualified workers. Sec-

CSCW 2017, February 25–March 1, 2017, Portland, OR, USA.

http://dx.doi.org/10.1145/2998181.2998249

ond, one might use experts or community leaders to focus the efforts of the community—to search in specific parts of the haystack (to license our analogy)—for example, in developing short stories [19], writing customer reviews [11], and idea generation [8]. This latter approach efficiently uses a given pool of workers by better directing their limited efforts, but requires experts to 1) shepherd different community members through tasks and 2) to know apriori (or to be able to rapidly screen) what makes submissions high quality. Those conditions may not be scalable or possible.

Leveraging Machines

Another common approach attempts to fully automate quality assessment through numerical models. The most common input—and the one this paper considers—uses the submission's text. For example, many studies on the relationship between superficial text features like word count and essay scores have found a positive relationship between the two [6]. In [20], Kobrin *et al.* discuss the relationship between length of response and scores on the SAT Essay and show that 39% of variance in essay scores are explained by the number of words used. Other similar studies where superficial features are predictive of quality ratings by experts are found in funding proposals [2] and resource quality for educational digital libraries [4].

In [17], the authors study the Webby awards dataset to judge good website designs and conclude that superficial metrics like number of fonts and word count are capable of predicting experts judgements. They provide two possible explanations for this behavior. First is a possible causal relationship between superficial metrics and deeper aspects of information architecture. The second explanation assumes that high quality in superficial attributes is generally accompanied by high quality in all aspects of a work. In other words, text features correlate well with quality work overall, and hence are good predictors even if they are not necessarily causal.

Different groups have also tried to predict whether or not an article will be featured on Wikipedia by using features such as complex words and readability indexes [39]. They found that simply predicting that articles longer than 2000 words will be featured achieves 96.3% accuracy. Computational models of quality have also focused specifically on readability [12] and coherence [16]. These types of measures have been used as features to train supervised machine learning models to predict human readers judgement [33] or in tasks such as demonstrating that discourse relations are strongly associated with the perceived quality of text [26].

Combining Humans and Machines

A third approach—and the one this paper adopts—is to combine the capabilities of humans and machines to play to their respective strengths. For example, Chan *et al.* [7] find that combining machine and human idea suggestions improves idea generation overall, compared to just human suggestions. Likewise, Siangliulue *et al.* [37] first use human crowds to compare pairs of idea, but then use metric-learning algorithms on that comparison data to project ideas into a 2D plane for exploration. They find that doing so allows people to find more diverse ideas. While that work addressed diversity, this work addresses quality estimation. Fuge *et al.* [13] find that combining content features and human rating recommends better design methods than either aspect independently. Lastly, a more distant, but related field is in using human feedback to guide machine learning techniques. A representative example is *Flock* by Cheng and Bernstein [9], which uses humans to help iteratively guide and label features for a classifier. While our work does not do this classifier introspection interactively, the approach used in Flock could be used to speed up and improve quality estimation in the future.

Scope and Contributions of this Paper

This paper proposes a complementary approach that combines the nuance of community participation with the scalability of automated filtering techniques: it builds a predictive model based on past submissions that can rank order new ideas by expected quality. In doing so, the model aims to provide higher-quality inspirations for community members, in turn supporting better collaborative work. Specifically, this paper answers two questions:

Q1 Compared to existing alternatives, such as comment sorting, how effectively do model-based rankings capture high-quality ideas, as judged by both whether they win a challenge and via human evaluation?

Q2 What aspects of submissions are highly influential at predicting quality scores, as judged by the ranking model?

To answer these questions, the paper analyzes community feedback, idea uniqueness, and text features from submissions on OpenIDEO [22] and then uses Gradient Boosted Trees to 1) understand which of those features predict high quality submissions, and 2) rank-order submissions by quality and compare that quality order to one of OpenIDEO's existing filters. We verify those results by comparing both discounted cumulative gain and human evaluation of ranked lists.

METHODOLOGY

Our approach to recommending a ranked list of ideas consists of three steps. First, we calculate a set of features representing different aspects of an idea like uniqueness, readability, coherence, and semantics. Second, we partition ideas by our proxy measure for quality: we assume that higher quality ideas are the ones that the OpenIDEO members advance to the Evaluation and Winner stages in the challenge—we explain these different stages below. Lastly, we divide the ideas into a training and test set and use Gradient Boosted Trees to predict the winning ideas. Given a trained classifier, we can apply it to unseen challenges and use the classification scores to rank order new ideas.

To evaluate that ranking, we use normalized Discounted Cumulative Gain (DCG), where relevance is 1 if the idea is a winner and 0 otherwise. (Alternatively, one can give relevance weight to *both* winning ideas as well as those that make it to the evaluation stage—doing so does not substantively change our below results.)

Dataset

We use 14 challenges summarized in Table 4 with 3918 ideas from completed challenges on OpenIDEO, an online design community where members design products, services, and experiences to solve broad social problems [22, 14]. OpenIDEO challenges have a problem description and stages—*e.g.*, Research, Ideas, Applause, Refinement, Evaluation, and Winners—to refine and select a small subset of winning ideas. During the *Ideas* stage, participants generate and view hundreds to thousands of design ideas; in practice, the number of submissions make exhaustive review (even of the titles) impossible—*e.g.*, for a medium-sized challenge of $\approx$ 500 ideas, it would take a person over 40 hours to read all idea entries). Our model aims to improve this stage by ordering or filtering ideas by quality, so that community members can review a manageable number of high-quality inspirations. During and after the Ideas stage, submitted ideas accrue views, applause, and comments as other participants provide feedback. Eventually a subset of ideas ($\approx 20-50$) advance to the *Evaluation stage*, and then a further subset (($\approx$ 10 – 15) advance to the *Winners stage* (see Table 4).

For each idea, we capture the following data at a common snapshot in time: 1) the text describing the idea, 2) the number and timestamp of any comments left on that idea, and 3) whether the idea advanced to the Evaluation or Winner stage. While there are various other data for each idea—the amount of applause, number of views, citation information, author statistics (location, site usage)—many of these features can change over time (even after the challenge has closed) and thus would be unfair indicators for a classifier focused on predicting unseen challenges. For example, winning ideas, given added publicity on the site after the fact, (expectedly) receive heavy view and applause once the challenge ends. For our analysis, we only use idea features that remain essentially static after once ideas enter the evaluation stage.

Idea Features

We use many features which may indicate the quality of an idea, broadly divided into the following groups:

Community Feedback These features indicate the response an idea receives from the OpenIDEO community. For example, a large number of comments received by an idea (prior to the Evaluation or Winning stage) indicates that the community is interested in the idea. Applause and view count, while valid measures of interest, are biased because 1) applause and views for winning ideas accrue over time as they proceeded to subsequent stages while other ideas do not (complicating post-challenge analysis since we do not have applause or views over time), and 2) OpenIDEO allows sorting by applause and views exacerbating the rich-get-richer problem and inflating low-effort measures like applause and views. For comment counts, winning ideas may accrue many comments congratulating them on their success. However, we estimated the evaluation stage's start date for each challenge and counted only comments before it using their timestamp. We refer to this modified quality feature as comment count. Figure 1 shows the box plot of comment count for 1) winning ideas, 2) evaluated ideas that did not win, and 3) all other ideas. As expected, on average winning ideas received more comments even before the announcement of evaluation stage results indicating that the community found them more interesting.

Author location OpenIDEO challenges received submissions from 87 countries. We found that 42% submissions come from the United States, which is understandable as OpenIDEO is a US based company. While a large proportion of submissions choose not to indicate the home country, English speaking countries dominated submissions. We did not find statistical bias for any country in choosing winners or evaluation stage ideas.

Text Descriptors We calculated a set of 22 surface descriptive text features using the Python readability package[1], including features like word or paragraph complexity, *etc.*[2] Figures 2, 3, and 4 show box plots for three text descriptors that (as we show later) are instrumental in successfully classifying winners: long words, sentences and vocabulary size. Each box plot shows the distribution for 1) winning ideas, 2) evaluated ideas that did not win, and 3) all other ideas.

Text Readability Readability is what makes some texts easier to read than others. We use the following readability measures—ARI, Coleman-Liau, Flesch Reading Ease, Gunning Fog Index, Kincaid, LIX, RIX, SMOG Index [31]. The Python readability package was also used to calculate these eight measures.

Text Coherence Coh-metrix is a computational tool which analyzes text for cohesion using 108 features mapped to five principal components: Narrativity, Deep cohesion, Referential cohesion, Syntactic Simplicity and Word Concreteness [28]. These features essentially measure how well an idea is narrated, the degree to which it contains connectives and conceptual links, how well ideas and words overlap across sentences, usage of fewer words and simple structure, and whether the text evokes mental images. We calculated these features using the online Coh-Metrix tool [3] [16] and observed that, in general, descriptions of OpenIDEO idea were less narrative but were well connected (high deep cohesion). Surprisingly, as we show below, coherence did not strongly influence our model's prediction of winners.

Text Semantics In addition to Coh-metrix (Coherence), we also use the Linguistic Inquiry and Word Count (LIWC) tool [32]. LIWC compares text to a dictionary that identifies which words are associated with psychologically-relevant categories, such as positive and negative emotions,

[1] https://pypi.python.org/pypi/readability/0.1

[2] The entire set of readability features are: the number of articles, number of auxillary verbs, number of characters, characters per word, number of complex words, number of conjunctions, number of interrogative words, number of long words with more than 7 characters, nominalization, number of paragraphs, number of preposition, number of pronouns, number of sentences, sentences per paragraph, subordination, syllables per word, number of syllables, to be verbs, type token ratio, number of words, words per sentence and size of vocabulary.

[3] http://tool.cohmetrix.com/

anger, sadness, *etc.*A complete list of the 93 features measured by LIWC is available at the LIWC website.[4] These capture higher-level semantics regarding the content and tone of the ideas. LIWC features have been used in a wide array of application areas ranging from predicting student course performance [34], identifying sarcasm on Twitter [15] to web-based depression treatment [10].

Idea Uniqueness So far, the above features calculated properties of an idea by itself. However, realistically the perceived quality of an idea may also depend on how it compares to other ideas within a challenge. Representativeness measures how similar the idea is to all other ideas in the collection. The assumption is that ideas which are unique to an existing set of ideas are more likely to have higher perceived quality compared to ideas which are similar to each other [1]. A common way to measure text similarity is through network models like TextRank [29], Graph approaches [24], and sub-modular functions [25]. We calculated representativeness by applying PageRank to cosine similarity matrix between idea topic proportions (similar to TextRank [29]). The idea topic proportions were estimated using Latent Dirichlet Allocation. The representativeness values were calculated for each challenge separately and then normalized due to different sizes of challenges. Figure 5 shows the box plot of normalized representativeness metric, where winning ideas and evaluation stage ideas are more unique.

We purposefully select only those features which can be directly estimated for new challenges and ideas, since in practice new submissions will not have applause, views, or other such history-dependent measures. Features like unigrams, bigrams, and topic proportions were purposefully not included in this analysis.

For example, new challenges may use a vocabulary which is domain specific and may not completely overlap with trained model vocabulary. This was the reason we did not add unigrams or TF-IDF in our analysis. Secondly, adding such features leads to a large increase in the input dimensions (3488 features compared to 319 features finally used). For brevity, we did not report classification results with TF-IDF or topic proportion features added, but they did not provide any significant improvement in classification performance.

Classification

The 14 challenges in our dataset had 3918 ideas with 3.5% of the ideas declared as winners. To address this class imbalance, we used MATLAB's RUSBoost algorithm, which under/over-samples data to balance classes [35]. The method was selected after comparing most standard classification methods including decision trees, logistic regression classifiers, discriminant analysis, support vector machines, nearest neighbour classifiers and ensemble classifiers like Boosted and Bagged trees. All methods were tested on various combination of test challenges and RUSBoost algorithm was consistently found to be better at classifying winners.

[4] http://liwc.wpengine.com/compare-dictionaries/

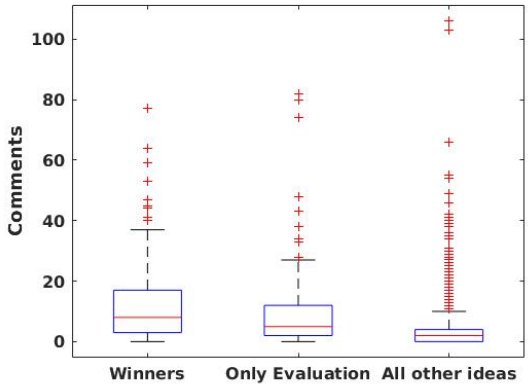

Figure 1. Comment count distribution. On average, winning ideas received 12 comments at the end of first stage compared to 9 comments for evaluation stage ideas and only 3 comments in initial stage ideas

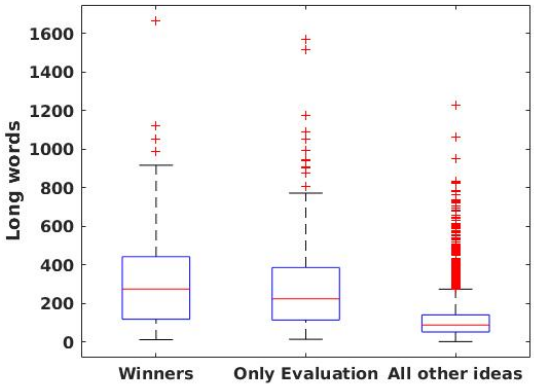

Figure 2. Long words distribution. On average, winning ideas contain 332 long words compared to 296 in evaluation stage ideas and 116 in initial stage ideas

We divided the dataset into 11 challenges for training and 3 challenges for testing and used 5-fold cross-validation. Note that we did not split ideas within a challenge between training and testing data but kept completely new challenges as the test set. An alternate approach would be to randomly split ideas from the bucket of all 3918 ideas, however such splitting will give artificially higher classification performance due to some challenge specific properties being manifested in the training set. Instead, we chose the more rigorous and realistic setting of only testing on completely unseen challenges.

Discounted Cumulative Gain

Normalized discounted cumulative gain (nDCG) measures the performance of a recommendation system based on the graded relevance of the recommended entities [18]. It varies from 0 to 1, with 1 representing the ideal ranking. This metric is commonly used in information retrieval to evaluate the performance of recommender systems. If k is the maximum number of entities that can be recommended, then DCG is

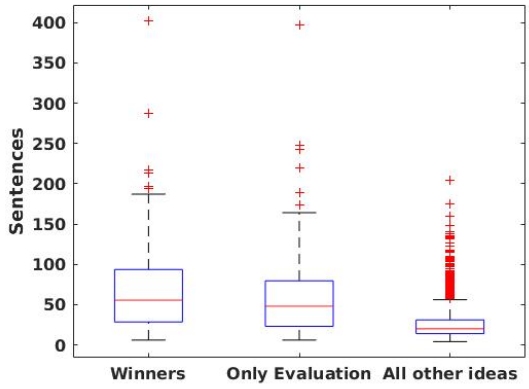

Figure 3. Distribution of number of sentences. On average, winning ideas contained 70 sentences compared to 60 sentences for evaluation stage ideas and only 26 sentences for initial stage ideas

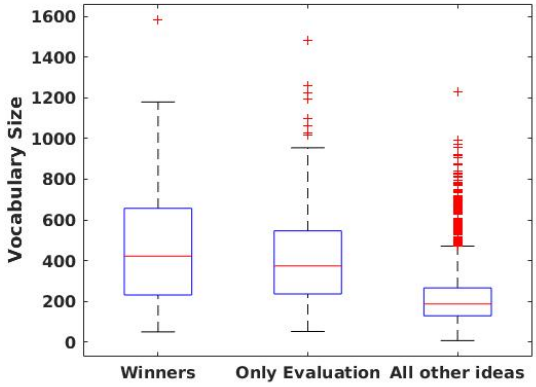

Figure 4. Distribution of vocabulary size. On average, winning ideas used a large vocabulary of 471 unique words compared to 427 words for evaluation stage ideas and only 215 words for initial stage ideas

given by:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{1}$$

$IDCG_k$ is defined as the maximum possible (ideal) DCG for a given set of ideas. Hence normalized DCG is given by:

$$nDCG_k = \frac{DCG_k}{IDCG_k} \tag{2}$$

We evaluate our methodology using two different definitions of relevance defined in Equation 3 and 4.

$$rel_i = \begin{cases} 1, & \text{if Idea } i \text{ is Winner} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

To get an intuitive understanding of DCG_k, consider the following example. Assume that a challenge has total 5 winners and that we get two ranked lists of 10 ideas each using method A and method B. List 1 is $[1, 1, 0, 1, 0, 0, 0, 0, 0, 1]$ where 1 indicates if the idea recommended is winning idea and 0 is a non-winning idea. List 2 is $[0, 1, 0, 1, 0, 0, 0, 0, 1, 1]$. While both lists have a total four winners, List 1 has more winners towards the start of the list. Using Equation 2, DCG_{10} for List 1 equals 2.35 and DCG_{10} for List 2 equals 1.65. Here, an ideal list will be [1, 1, 1, 1, 1, 0, 0, 0, 0, 0] where all winners are at the start of the list and $IDCG_{10}$ is 2.9485. Hence, $nDCG_{10}$ for List 1 is 0.92 while for List 2 is 0.64. Using this metric, Method A will be a preferred method as it provides more winners early on. We use *nDCGk* below to compare different methods for recommending ideas. Later, it is also shown that nDCG is more robust to inter-rater differences.

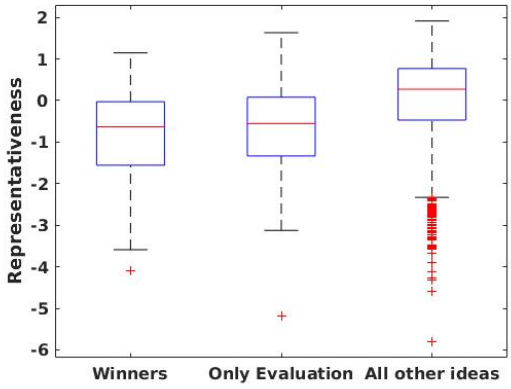

Figure 5. Distribution of normalized representativeness. Negative values show unique ideas while positive values are for ideas which are similar to most other ideas within the challenge. On average, winning ideas are more unique.

RESULTS AND DISCUSSION

We begin our results by presenting general classification accuracy of the model in predicting winners. Discussions on important features show how feature importance can provide cues to underlying processes in winner selection. Having demonstrated the classifier performance, we show superior *nDCG* performance for our proposed model over current OpenIDEO ranking methods and finally superior performance on human evaluated idea quality for our method over current OpenIDEO ranking methods is shown.

Predicting Winners

We used the following set of 319 features for the classification problem—LIWC (93 features), Coh Metrix (107 features), Text Descriptive (22 features), Readability (8 features), Author Location (88 binary features), Normalized Representativeness (1 feature) and Corrected Comment count (1 feature)

For the first part of analysis, challenges 3, 5, and 13 in Table 4 were randomly chosen for testing and remaining were used as training data. This led to 3442 ideas in training set and 550 ideas in test set with 26 winners in the test set. The results of the classification model to predict winners from all ideas are summarized in Table 1. The method captures 20 winners out of 26 true winners from the three test challenges achieving a recall value of 0.77 and precision of 0.19.

To further investigate the performance of the classifier and key factors which were instrumental in identifying the winners, we looked into the predictor importance. Predictor importance was estimated for trees by summing changes in the mean squared error due to splits on every predictor and dividing the sum by the number of branch nodes. This sum is taken over best splits found at each branch node using the "predictorImportance" MATLAB function.

Comments, Sentences, and Long words were found to be the most important features in classifying winners as shown in Fig. 6. This is also evident from the box plots shown before, where winners were easily distinguishable from other ideas for these features. Surprisingly, out of 319 features, 298 features had zero predictor importance in classification while significant contribution was made by only three features. Text coherence measured by Coh-metrix, semantic meaning measured by LIWC and most other surface readability measure had little contribution in winner and evaluation ideas identification.

Among the top three features, comments is an indicator of feedback received by community, showing winning is positively correlated with amount of feedback from community members. The remaining important predictive text descriptor features like sentences, long words, and word types were all strongly correlated with each other and dependent on document length. For instance, sentence count was strongly correlated (Pearson's linear correlation coefficient > 0.8) with number of characters, number of complex words, number of long words, number of syllables, number of to be verbs, number of words, and size of vocabulary. Count of long words strongly correlated (Pearson's linear correlation coefficient > 0.8) with count of auxillary verbs, characters, complex words, nominalization, sentences, syllables, to be verbs, words, and size of vocabulary. In essence, these correlated features substitute for and predict one another, with the classifier using those features to encode overall idea length.

Overall, winning ideas across OpenIDEO challenges get more user feedback and are longer documents with many sentences, longer words, and a larger vocabulary. They also tend to have unique topics compared to other ideas. While increased community engagement and uniqueness both seem like understandable winning qualities, the impact of text surface characteristics like length on winning ideas seems counter intuitive. Why should length help predict winners more than qualities like writing coherence? To understand this outcome further, we looked at the following questions:

- Are winning ideas actually good quality?
- Does collaboration lead to longer ideas and winning?

Are winning ideas good quality?

Our approach assumes that ideas that reach the latter Evaluation and Winner stages have higher quality on average than those that do not. Is this assumption reasonable? For example, if reviewers are using "lazy" shortcuts like length (independent of content) to select winners, then high-quality ideas may not win. For OpenIDEO specifically, many mechanisms encourage winning submissions to have high quality: 1) during the evaluation stage, the community members rate ideas using a common rubric with criteria tied to challenge requirements; 2) a separate evaluation panel, which includes the challenge sponsor, discusses and selects the winning ideas from those evaluated by the community; and 3) certain sponsors may award funding to select winning ideas, increasing the panel's incentive to select the highest quality ideas addressing the challenge brief. These processes help winning ideas become high-quality, and vice versa. This may not be the case for other types of collaborative communities or crowd work, especially in cases where reviewers are rewarded for volume of work (such as in Amazon's Mechanical Turk), and special quality safe-guards should be put in place if someone wanted to apply our approach to those communities.

Does collaboration lead to longer ideas and winning?

Even if winning ideas are high-quality, does an idea's length really increase the chance of winning? In other words, is length a likely causal factor or is it merely correlated with an unseen latent factor, like the amount of time a user spends on refining the idea? Since our classifier model is purely predictive, it cannot measure such causal effects directly. To shed light on this question, however, we can consider two qualitative examples where winning and non-winning ideas had otherwise similar features except for length or comment count.

Similar length

For the first case, we compare two ideas—the first did not win while the second did—that had similar length and uniqueness, but differed in their comment counts. These ideas came from challenge five, which focused on increasing health outcomes (such as fitness, nutrition, *etc.*) in sedentary workplaces.[5] The two ideas and their summary sentences are:

1. Healthy Trucker Alliance Spreads the Word, Connects Drivers: Health statistics among truckers have been alarming, but more and more truckers are adopting positive changes. If drivers committing to healthy change displayed a standard placard logo at the tail of their rig, this might help the trend "go viral".[6]

2. Climbing Mount Everest, one step at a time: Let's enroll organisations in a fitness challenge to climb the equivalent height of Mt. Everest (and other similar challenges), a total of 58,070 steps, by walking, running, biking, taking the stairs. Also, add a competitive and fundraising element.[7]

When qualitatively comparing the commenting behavior between the two ideas, we noticed two possible reasons behind the second idea progressing: 1) the impact of community engagement, and 2) time spent on task.

First, the winning idea (Everest) initially attracted more community feedback comments (17 comments) compared to the

[5] https://challenges.openideo.com/challenge/well-work/brief

[6] https://challenges.openideo.com/challenge/well-work/concepting/healthy-trucker-alliance-spreads-the-word-connects-drivers/

[7] https://challenges.openideo.com/challenge/well-work/concepting/climbing-mount-everest-one-step-at-a-time/

Healthy Trucker idea (4 comments). More importantly, the author continuously engaged the community by replying to comments and updating the idea to incorporate their suggestions. This led to a final tally of 86 total comments for the idea, with 35 comments from the author himself. While OpenIDEO does not provide revision history, we noticed that the author's replies in the comments often mentioned updates and additions he made to the idea. This behavior likely increased the document length as the challenge progressed. Comparing this behavior to winning ideas from different challenges, we found similar engagement patterns wherein the authors engaged the community and updated their idea. Ideas which were long but received no community engagement, either due to lack of response from the author or lack of interest by the community, did not progress to further stages. This provides a possible explanation behind winning ideas being generally longer with more comment counts: increased engagement causes updates, which in turn increase length.

Second, long ideas can indicate time spent on task by the author (*i.e.*, effort) regardless of community feedback. A writer who spends more time refining his or her idea may, by virtue of covering more details, result in longer documents overall. Without complete revision history, however, this time on task is difficult and complicated to measure reliably. Time on task clearly increased when authors responded to comments and updated their idea, however we cannot readily identify the inverse case (lots of effort but few comments) without making many (potentially incorrect) assumptions about how that time manifests itself in the submission (*e.g.*, in better grammar, more uniform coverage of challenge requirements, *etc.*).

Similar comment count

To complement the above, we also compared a sets of ideas that had similar comment counts, but differed in length. Specifically, we looked at two ideas "On your way home"[8] and "The ultimative fitness software"[9] from the above fitness challenge. The 'Fitness software' idea did not reach evaluation or winning stage while 'On your way home' did. While both had same number of comments before the evaluation stage, the non-winning idea was actually **longer** (109 sentences) than the winning idea (70 sentences), which at first glance seems counter to our above results. The difference lies in how the authors used that community feedback.

The winning idea's author regularly revised the idea to incorporate community suggestions as indicated by comment replies (20 comments by author) (even changing the title to reflect the updates). In contrast, for the non-winning idea (fitness software), the idea's 'last modified date' occurred *before* the first comment, indicating the idea was not revised based on the feedback. While anecdotal, this again demonstrates that although document length and comments are important features for predicting winners, they might be caused by underlying factors like community engagement and idea revision, rather than purely time on task. In practice, community engagement and authors effort (through time-on-task) likely interact to cause advancement to the winning stage (though our current model cannot prove this causality). Hence, comments lead to co-creation, which lead to longer and better ideas. We also explored interaction effects between comments and sentences by isolating features, but no significant effect was found on the model performance. Another alternative explanation is large number of comments may increase the visibility of an idea, leading to preferential attachment. Unfortunately, due to lack of time series data we did not study preferential attachment by members.

[8] https://challenges.openideo.com/challenge/well-work/concepting/on-your-way-home

[9] https://challenges.openideo.com/challenge/well-work/concepting/the-ultimative-fitness-software-

Predictive performance when removing features

Using all features, the trained classifier achieves an 81% reduction in total ideas that communities need to process while still captureing 77% of the winning ideas for new challenges. One can use this classifier to filter ideas by quality (as measured by likelihood of winning). The results hold if we shift the goal from predicting winners to predicting ideas that reach the evaluation stage: Table 2 show that we achieve recall of 0.68 and precision of 0.39 for test challenges, while the important predictors remain Comments, Sentences, Long words, Size of vocabulary and Representativeness. While we only present one particular set of test challenges for clarity, these results hold across for different training and test challenges. First, however, we compare how the predictive model performance changes as we remove high-importance features (such as comment count and length), causing the classifier to differentiate winning versus non-winning submissions among different factors.

First, we eliminated comment count as a quality feature. The model still achieved a reduction of 75% in total ideas and captured 77% winning ideas for test set with challenges 3, 5 and 13. Second, we also eliminated any features directly related to document length, such as number of sentences, *etc.*We kept any text features that were normalized by the number of words, since this should minimize dependence on length. Under these conditions, the precision dropped to 0.12 while recall was 0.81 for test challenges. Important predictors are shown in Fig. 7. Given no knowledge of comment count or length, the classifier differentiated ideas by whether they were unique and easy to read.

Under this reduced model, idea uniqueness, as measured by text representativeness, was the most important predictor. As one would expect, winning ideas are generally not similar to most other ideas. This factor of uniqueness may also have attracted community attention to such ideas.

Next, the model selected two text coherence features relating to lexical diversity to differentiate winners. Lexical diversity is measured by two features (type-token ratio and LD-VOC) that essentially encode whether the document uses similar words throughout the document. Text with low lexical diversity are less complex and easier to read (all other things being equal). The model also used LIWC features for informal language such as netspeak (words like 'btw', 'lol', 'thx') or a lower percentage of common dictionary words (*i.e.*, informal words or unique words specific to an idea like place or person pronouns). Some punctuation features like colons and apos-

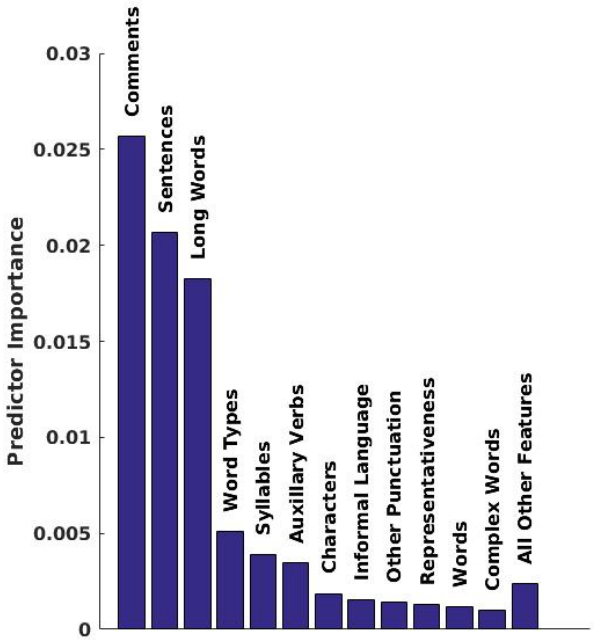

Figure 6. Feature importance in the model for winner prediction in Table 1. The number of comments, sentences, and long words are the most predominant features while 307 of 319 features have little importance.

trophes (normalized by document length) were also important. While this analysis showed easier language and unique ideas were more likely to win, we believe that some of these features might be indirectly affected by document length. For example, usage of colon was mostly in long documents to provide hyperlinks and lists of action items. This can be verified from the previously discussed winning idea, 'Climbing Mount Everest, one step at a time', which had a high score on colons, but mostly due to listing items and providing hyperlinks. Shorter documents might not have such features. We should ultimately take the above observations with a grain of salt, however, because although these results seem reasonable, those features do not meaningfully alter the classification performance when compared to features like comment count, length, or uniqueness.

While predicting winners gives useful insight into the model, our main goal is to rank order ideas so that design community participants can gain inspiration from such ideas during a new challenge. We explore this facet in the next section by using ranked list metrics. In the rest of the paper, we use the full set of features to rank ideas.

	Prediction			
	Validation		Testing	
	0	1	0	1
Non-Winner	2773	481	441	83
True Winner	44	70	6	20

Table 1. Confusion matrix for Validation and Test data for Winners

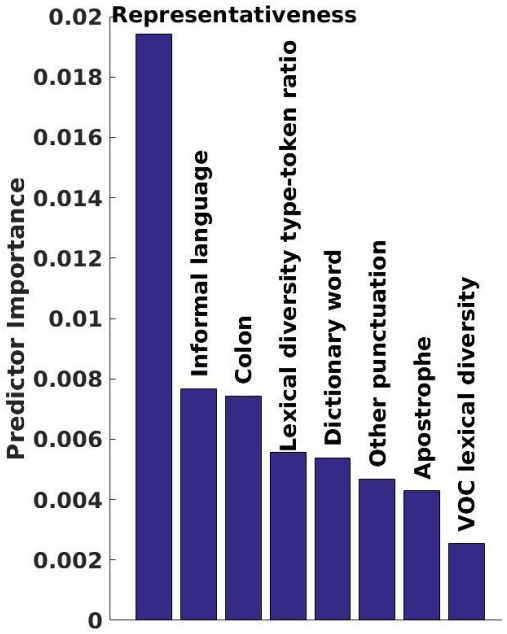

Figure 7. Feature importance in the reduced model (without comment or document length features).

	Prediction			
	Validation		Testing	
	0	1	0	1
Non-Winner	2681	415	427	63
True Winner	111	161	19	41

Table 2. Confusion matrix for Validation and Test data for Evaluations

Ranking Ideas

To ensure that prediction performance holds across various test challenges, we ran our model over every possible permutation of 3 test and 11 training challenges—a total 364 different train/test combinations. A more realistic performance measure is the rank order of ideas by quality and not predicting winners. For this we use $nDCG_k$, as defined before in Equation 1.

To compare the classification results, we obtain two lists. First list uses the default OpenIDEO ranking of sorting ideas by comment count—this acts as our baseline. For the second list, we rank order all ideas according to our model's classification score. Here, classification score is the probability of observing an instance of winning class in the leaves of the Gradient Boosted Tree. We use $nDCG_p$ to compare performance for the two lists using relevance defined in Equation 3. An ideal list for a challenge using this relevance should be rank ordered to contain all winners first, followed by all other ideas. Such a list will have maximum DCG and will be used to normalize DCG for comparison across challenges using Equation 2.

We calculate the $nDCG_{p(i,j)}$ for each challenge in training and test data for each of the 364 permutations. This gives a ma-

trix of 364 × 14 DCG values. Here $p(i, j)$ is the size of the challenge j for model i for a ranking of all ideas. Figure 8 shows the comparisons for 364 different permutations for training and test challenges using box plots. Here, $nDCG_p$ is first calculated for each challenge in the training set and then the challenges are averaged to calculate the average model $nDCG_p$ for training and test set. On average the classifier provides significantly better mean $nDCG$ values compared to sorting by comments.[10]

On browsing the OpenIDEO challenges, the first page of any challenge shows 21 ideas irrespective of the size of the challenge. To check how the two lists compared in providing ranked recommendations only on the first page we also calculated $nDCG_{21}$. This essentially captures how many winners are captured on the first page of recommendations by a list. Figure 9 shows the corresponding results, with the classifier performing better than ranked comments.[11] Using different values of list length does not substantively change the results.

If we consider both winning ideas *and* evaluated ideas to represent quality, we can use the alternative measure of relevance for DCG defined in Equation 4. We used 0.4 as relevance of idea reaching evaluation stage. The value was estimated to by considering the proportion of ideas that reach evaluation stage compared to number of winners across all 14 challenges.

$$rel_i = \begin{cases} 1, & \text{if Idea } i \text{ is Winner} \\ 0.4, & \text{if Idea } i \text{ only reached Evaluation} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Figure 10 shows the comparison of the two lists for $nDCG_{21}$ using both winners and evaluation ideas. Again, even though the classification model was trained to predict winners only, it also captures more evaluation stage ideas in first 21 entries for different challenges compared to sorting by comment counts[12]. This indicates that the model is capturing some predictive power for quality (again assuming that evaluated ideas have higher quality on average than non-evaluated ideas).

So far, we have averaged the performance for the test and training sets for all 364 cases. However, the classifier performance naturally varies across different challenges. Figure 11 shows the $nDCG$ values using relevance from Equation 4 for each challenge (both test and train) for all the 364 models. Winners for challenges 2, 3, 4, and 13 are consistently easy to predict while some challenges like 1, 12, and 14 are difficult to predict. The $nDCG_{21}$ values for comment counts are also shown for each challenge using the diamond marker. We found that on average, challenges 12 and 14 had unusually small document lengths (less than 20 sentences) and less comment count compared to other challenges. Similarly, challenge 1 had very high sentence count (on average more than 40 sentences) compared to other challenges. Hence, the classifier found it difficult to predict winners in these outlier challenges. Understanding what differentiates quality across challenges, factors for lesser community engagement in some

[10]two-sample t-test, $N = 364$, $\Delta DCG = 0.073$, $p = 4.51 \times 10^{-61}$
[11]two-sample t-test, $N = 364$, $\Delta DCG = 0.042$, $p = 5.63 \times 10^{-09}$
[12]two-sample t-test, $N = 364$, $\Delta DCG = 0.020$, $p = 0.005$

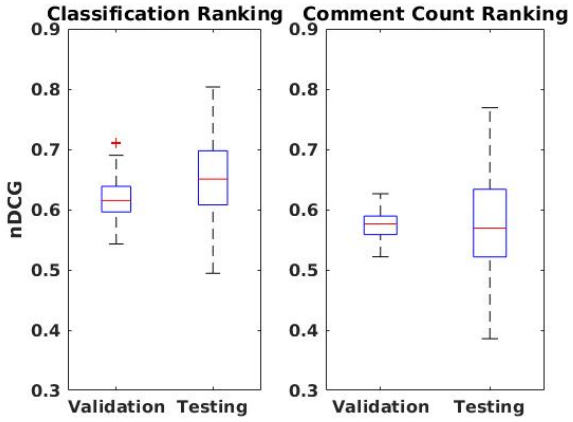

Figure 8. Mean DCG_{all} of all 364 cases for Validation and Testing datasets to compare classifier and comment count lists. On average, the classifier gets a higher $nDCG$

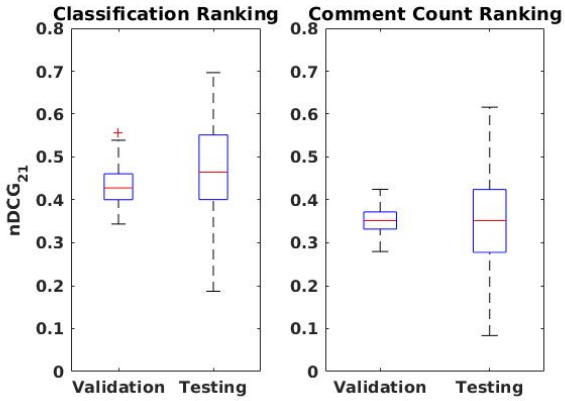

Figure 9. Mean DCG_{21} of all 364 cases for Validation and Testing and comparison of classification and comment count lists. On average, the classifier gets more winning ideas in the first 21 recommendations across challenges for different models

challenges and shorter ideas would be an interesting avenue for future work which our limited set of features did not capture.

Human Evaluation

So far, we have assumed that winning ideas are a metric of high quality and used it as the relevance measure for an idea. However, to verify this assumption, we tested whether actual humans thought the higher $nDCG$ lists had higher quality overall. Four evaluators were given links to two ranked list of 10 OpenIDEO ideas each from challenge 5 ("How might we create healthy communities within and beyond the workplace?"). Two of these evaluators were professors and other two were graduate students. Three of the four evaluators had previous experience with design ideas typical of those on OpenIDEO. List 1 contained 10 ideas sorted by comments—the default OpenIDEO sorting order—while List 2 was sorted by classification scores for model trained using test cases 3, 5, and 13 discussed earlier.

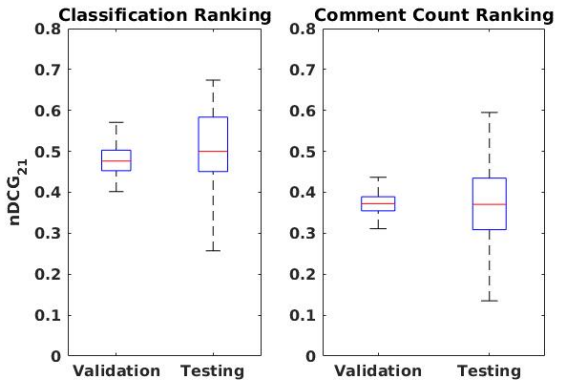

Figure 10. Mean DCG_{21} with relevance including evaluation ideas for all 364 cases for Validation and Testing and comparison of classification and comment count lists. On average, the classifier recommends more winning and evaluation stage ideas in the first 21 recommendations across challenges for different models

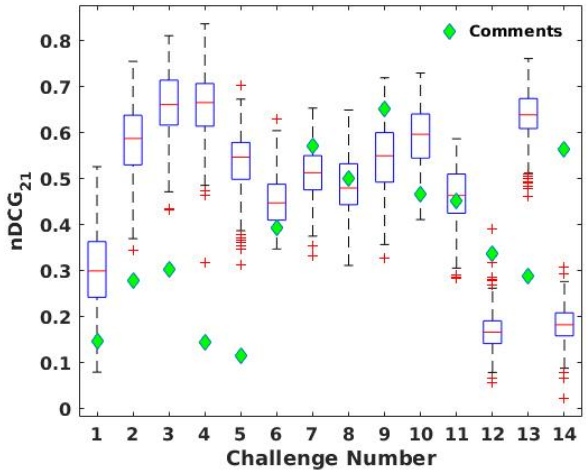

Figure 11. Mean DCG_{21} for each challenge using relevance formulation from Equation 4 showing some challenges are always difficult to predict. The comment count DCG_{21} is also shown for the challenges by diamond markers.

Each evaluator scored ideas on a Likert scale of 1 to 5 for the quality of the idea as it relates to the challenge brief. We provided them with the challenge brief and explicit instructions to ignore applauds, views, comments, or challenge status while reviewing ideas. On average, evaluators took 40 minutes to rate the two lists.

The average rating for List 1 (comment sorting) was 3.08 while for List 2 (our model) was 3.50. Comparing the distributions of raw quality ratings across all four evaluators, List 1 and 2 did not differ significantly.[13] However, rather than averages, we are more interested in whether the quality rankings improve. To compare $nDCG_{10}$ the Likert ratings were scaled between 0 to 1 and relevance defined as:

$$rel_i = \text{Scaled Likert rating for idea at position i} \quad (5)$$

Table lists the corresponding $nDCG_{10}$ values for each evaluator. The classifier (List 2) consistently produced a better ranking than comment count sorting (List 1). List 2 received better average ratings and also placed higher rated ideas further up the list compared to List 1.[14] While the classifier was trained to predict only winners, the corresponding evaluator ratings illustrate that it likely captures an independent subjective measure of quality, at least compared to the current OpenIDEO default of ordering by comment counts.

The subjective nature of ordinal ratings meant that inter-rater reliability (IRR) tests (*e.g.*, Linear weighed Cohen's Kappa 0.01 and observed agreement 0.72 between 1 and 4) showed only slight agreement between raters. However, such IRR tests compare absolute magnitudes of ratings which naturally vary by rater (*e.g.*, two raters may have different standards for what deserves a '5' rating or a '1' rating, even if they agree on which idea is best or worst). Comparing the evaluator agreement among nDCG values accounts for this by only comparing rank orders. The rank orders for three evaluators (1, 3, and 4) were largely consistent, while one evaluator (2—the one without experience rating design ideas) disagreed with the rankings. With the current sample size it is difficult to read too deeply into these comparisons. However, they do provide some evidence that winning ideas do match human judgements of quality, and that the classifier outperformed standard comment sorting.

List	Evaluator 1	2	3	4	Mean DCG
Comment Count	0.78	0.84	0.72	0.71	0.76
Classifier	0.90	0.80	0.94	0.92	0.89

Table 3. Normalized Discounted Cumulative Gain using evaluator quality ratings as measure of relevance

Use and Limitations

Overall, our results point to an automated means to complement existing crowd-based methods of idea selection. By combining the text content of the idea with quality and uniqueness, our approach can be used directly after idea submission—avoiding the cold-start problem. Likewise, it avoids the rich-get-richer problem because it does not rely on solely on instantaneous popularity of particular ideas within a challenge. Instead, our approach builds a joint model over winning ideas across challenges. Finally, it can provide initial quality ratings to all submissions, regardless of crowd size, negating (or at least limiting) the sparsity problem.

Our proposed method is limited in several important respects. First, it assumes that quality is approximated via ideas that move to progressive stages of a crowd selection process. This creates a chicken and egg problem: in order to help a crowd select good ideas—*i.e.*, to avoid the mentioned problems—it first relies on the crowd having selected reasonably good ideas! In practice, we believe that existing voting dynamics are sufficient to move at least *some* good ideas move forward, providing a reasonable basis for the model. Combining both human- and machine-based approaches for estimating quality should create more value than the sum of their parts.

[13] two-sample t-test, $N = 4$, $\Delta DCG = 0.42$, $p = 0.13$

[14] two-sample t-test, $N = 4$, $\Delta DCG = 0.13$, $p = 0.023$

Second, we were surprised that higher level features, such as coherence and semantics, did not play a more important role in defining good ideas. Future, higher-level feature representations—*e.g.*, over topical content or functional structures—might prove more fruitful.

Third, the winning ideas within an OpenIDEO challenge are not based solely on quality, and include aspects such as diversity, which our proposed model does not attempt to describe. Combining this approach with other formal models of diverse idea selection [37] would likely improve results.

CONCLUSION

This paper presented an approach for estimating the quality of ideas through a classifier than learns to differentiate winning ideas from non-winning ideas. We found that basic low-level text features, such as number of sentences and long words, when combined with the representativeness of a document (measured in topic space) and the number of comments provides the most useful ranking information; in contrast, features such as writing coherence, semantic meaning (as represented by LIWC) did not substantively improve rankings. We also found that our text-based ranking model improves the ranking performance (as measured through Discounted Cumulative Gain) when compared to OpenIDEO's existing comment ranking. The human evaluations bear similar results.

These results indicate that text-based quality models of submissions would complement existing approaches to understanding and organizing submissions created by collaborative online workers. The model helps those large groups of workers build productively upon the good ideas of others without becoming burdened by the increasing quantity of submissions. While our data and examples focused on online design communities, these results could extend to larger collaborative working groups provided those groups use a procedure for robustly measuring the quality of submissions (an idea easier said than done when dealing with workers paid for quantity over quality).

Our proposed model could combine the nuanced, subjective assessments of crowd-driven idea selection with the scalability of pre-trained classifiers. While neither approach by itself is likely to completely solve scalable idea selection, they hold more promise together than they do separately. By helping people identify the needles in the crowd haystack, such models could ultimately improve both the speed and effectiveness with which large groups of workers collaborate together on complex tasks.

ACKNOWLEDGMENTS

We thank Josefine Engel for help calculating some of the features used in the paper, and Dr. Vanessa Frias-Martinez for helpful discussions. We also thank the evaluators for participating in the survey. We also thank the reviewers for their insightful comments that significantly improved the quality and depth of the paper.

REFERENCES

1. Faez Ahmed, Mark Fuge, and Lev D. Gorbunov. 2016. Discovering diverse, high quality design ideas from a large corpus. In *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. ASME.
2. Daniel Berleant. 2000. Does typography affect proposal assessment? *Commun. ACM* 43, 8 (2000), 24–24.
3. Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2015. Soylent: A Word Processor with a Crowd Inside. *Commun. ACM* 58, 8 (July 2015), 85–94. DOI: http://dx.doi.org/10.1145/2791285
4. Steven Bethard, Philipp Wetzer, Kirsten Butcher, James H. Martin, and Tamara Sumner. 2009. Automatically Characterizing Resource Quality for Educational Digital Libraries. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '09)*. ACM, New York, NY, USA, 221–230. DOI:http://dx.doi.org/10.1145/1555400.1555436
5. Daren C. Brabham. 2008. Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. *Convergence: The International Journal of Research into New Media Technologies* 14, 1 (2008), 75–90. DOI:http://dx.doi.org/10.1177/1354856507084420
6. Jill Burstein, Martin Chodorow, and Claudia Leacock. 2004. Automated essay evaluation: The criterion online writing service. *AI Magazine* 25, 3 (2004), 27. http://dx.doi.org/10.1609/aimag.v25i3.1774
7. Joel Chan, Steven Dang, and Steven P. Dow. 2016a. Comparing Different Sensemaking Approaches for Large-Scale Ideation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2717–2728. DOI:http://dx.doi.org/10.1145/2858036.2858178
8. Joel Chan, Steven Dang, and Steven P. Dow. 2016b. Improving Crowd Innovation with Expert Facilitation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, New York, NY, USA, 1223–1235. DOI: http://dx.doi.org/10.1145/2818048.2820023
9. Justin Cheng and Michael S. Bernstein. 2015. Flock: Hybrid Crowd-Machine Learning Classifiers. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 600–611. DOI:http://dx.doi.org/10.1145/2675133.2675214
10. Rianne Van der Zanden, Keshia Curie, Monique Van Londen, Jeannet Kramer, Gerard Steen, and Pim Cuijpers. 2014. Web-based depression treatment: Associations of clients word use with adherence and outcome. *Journal of Affective Disorders* 160 (2014), 10 – 13. DOI: http://dx.doi.org/10.1016/j.jad.2014.01.005

Title	Number of ideas	Evaluation	Winners
1. How might we make low-income urban areas safer and more empowering for women and girls?	573	52	15
2. How might we inspire young people to cultivate their creative confidence?	608	22	9
3. How might we all maintain well-being and thrive as we age?	134	20	6
4. How might we gather information from hard-to-access areas to prevent mass violence against civilians?	166	17	6
5. How might we create healthy communities within and beyond the workplace?	240	20	10
6. How can we equip young people with the skills, information and opportunities to succeed in the world of work?	148	20	6
7. How might we support web entrepreneurs in launching and growing sustainable global businesses?	157	20	10
8. How might we design an accessible election experience for everyone?	154	20	11
9. How might we restore vibrancy in cities and regions facing economic decline?	326	20	11
10. How can technology help people working to uphold human rights in the face of unlawful detention?	165	16	9
11. How might we better connect food production and consumption?	606	20	10
12. How might we increase the number of registered bone marrow donors to help save more lives?	285	25	10
13. How might we improve maternal health with mobile technologies for low-income countries?	176	20	10
14. How can we raise kids awareness of the benefits of fresh food so they can make better choices?	180	40	17

Table 4. 14 Challenges incorporated in dataset showing the size of the challenge, number of winners and evaluation stage ideas

11. Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the Crowd Yields Better Work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 1013–1022. DOI: http://dx.doi.org/10.1145/2145204.2145355

12. WH DuBay. 2008. The principles of readability. 2004. *Costa Mesa: Impact Information* (2008), 77. DOI: http://dx.doi.org/10.1.1.91.4042

13. Mark Fuge, Bud Peters, and Alice Agogino. 2014a. Machine Learning Algorithms for Recommending Design Methods. *Journal of Mechanical Design* 136, 10 (18 Aug. 2014), 101103+. DOI: http://dx.doi.org/10.1115/1.4028102

14. Mark Fuge, Kevin Tee, Alice Agogino, and Nathan Maton. 2014b. Analysis of Collaborative Design Networks: A Case Study of OpenIDEO. *Journal of Computing and Information Science in Engineering* 14, 2 (March 2014), 021009+. DOI: http://dx.doi.org/10.1115/1.4026510

15. Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying Sarcasm in Twitter: A Closer Look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 581–586. http://dl.acm.org/citation.cfm?id=2002736.2002850

16. Arthur C. Graesser, Danielle S. McNamara, Max M. Louwerse, and Zhiqiang Cai. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers* 36, 2 (2004), 193–202. DOI: http://dx.doi.org/10.3758/BF03195564

17. Melody Y. Ivory, Rashmi R. Sinha, and Marti A. Hearst. 2001. Empirically Validated Web Page Design Metrics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 53–60. DOI: http://dx.doi.org/10.1145/365024.365035

18. Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 422–446. DOI: http://dx.doi.org/10.1145/582415.582418

19. Joy Kim, Justin Cheng, and Michael S. Bernstein. 2014. Ensemble: Exploring Complementary Strengths of Leaders and Crowds in Creative Collaboration. In *Proceedings of the 17th ACM Conference on Computer*

Supported Cooperative Work & Social Computing (CSCW '14). ACM, New York, NY, USA, 745–755. DOI:http://dx.doi.org/10.1145/2531602.2531638

20. Jennifer L Kobrin, Hui Deng, and Emily J Shaw. 2007. Does Quantity Equal Quality? The Relationship between Length of Response and Scores on the SAT Essay. *Journal of Applied Testing Technology* 8, 1 (2007), 1–15.
21. V. Kostakos. 2009. Is the Crowd's Wisdom Biased? A Quantitative Analysis of Three Online Communities. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, Vol. 4. 251–255. DOI:http://dx.doi.org/10.1109/CSE.2009.491
22. Karim Lakhani, Anne-Laure Fayard, Natalia Levina, and Stephanie Healy Pokrywa. 2012. OpenIDEO. *Harvard Business School Technology & Operations Mgt. Unit Case* 612-066 (2012).
23. Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. 2008. Addressing Cold-start Problem in Recommendation Systems. In *Proceedings of the 2Nd International Conference on Ubiquitous Information Management and Communication (ICUIMC '08)*. ACM, New York, NY, USA, 208–211. DOI: http://dx.doi.org/10.1145/1352793.1352837
24. Laura Langohr and others. 2014. *Methods for finding interesting nodes in weighted graphs*. Ph.D. Dissertation. University of Helsinki.
25. Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 510–520.
26. Annie Louis and Ani Nenkova. 2013. What Makes Writing Great? First Experiments on Article Quality Prediction in the Science Journalism Domain. *Transactions of the Association for Computational Linguistics* 1 (2013), 341–352.
27. Kurt Luther, Nathan Hahn, Steven P Dow, and Aniket Kittur. 2015. Crowdlines: Supporting Synthesis of Diverse Information Sources through Crowdsourced Outlines. In *Third AAAI Conference on Human Computation and Crowdsourcing*.
28. Danielle S McNamara, Arthur C Graesser, Philip M McCarthy, and Zhiqiang Cai. 2014. *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press.
29. Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of EMNLP 2004*, Dekang Lin and Dekai Wu (Eds.). Association for Computational Linguistics, Barcelona, Spain, 404–411. http://www.aclweb.org/anthology/W04-3252
30. Michael I Norton and Jeremy B Dann. 2011. Local motors: designed by the crowd, built by the customer. *Harvard Business School Marketing Unit Case* 510-062 (2011).
31. João Rafael de Moura Palotti, Guido Zuccon, and Allan Hanbury. 2015. The Influence of Pre-processing on the Estimation of Readability of Web Documents. (2015), 1763–1766. DOI: http://dx.doi.org/10.1145/2806416.2806613
32. James W. Pennebaker, Martha E. Francis, and Roger J. Booth. 2001. Linguistic inquiry and word count: LIWC 2001. (2001).
33. Emily Pitler and Ani Nenkova. 2008. Revisiting Readability: A Unified Framework for Predicting Text Quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 186–195. http://dl.acm.org/citation.cfm?id=1613715.1613742
34. Rebecca L. Robinson, Reanelle Navea, and William Ickes. 2013. Predicting Final Course Performance From Students Written Self-Introductions: A LIWC Analysis. *Journal of Language and Social Psychology* 32, 4 (2013), 469–479. DOI: http://dx.doi.org/10.1177/0261927X13476869
35. Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. 2010. RUSBoost: A hybrid approach to alleviating class imbalance. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 40, 1 (2010), 185–197.
36. Burr Settles and Steven Dow. 2013. Let's Get Together: The Formation and Success of Online Creative Collaborations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2009–2018. DOI:http://dx.doi.org/10.1145/2470654.2466266
37. Pao Siangliulue, Kenneth C. Arnold, Krzysztof Z. Gajos, and Steven P. Dow. 2015. Toward Collaborative Ideation at Scale: Leveraging Ideas from Others to Generate More Creative and Diverse Ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 937–945. DOI:http://dx.doi.org/10.1145/2675133.2675239
38. Yla R. Tausczik, Aniket Kittur, and Robert E. Kraut. 2014. Collaborative Problem Solving: A Study of MathOverflow. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 355–367. DOI: http://dx.doi.org/10.1145/2531602.2531690
39. Philipp Wetzler, Steven Bethard, Heather Leary, Kirsten Butcher, Soheil Danesh Bahreini, Jin Zhao, James H. Martin, and Tamara Sumner. 2013. Characterizing and Predicting the Multifaceted Nature of Quality in Educational Web Resources. *ACM Trans. Interact. Intell. Syst.* 3, 3, Article 15 (Oct. 2013), 25 pages. DOI: http://dx.doi.org/10.1145/2533670.2533673