Challenge AI's Mind: A Crowd System for Proactive AI Testing

Siwei Fu

Zhejiang Lab
1818 Wenyi West Road, Hangzhou, Zhejiang, China
Email: fusiwei339@gmail.com

Xiaotong Liu, Anbang Xu, Rama Akkiraju

IBM Research-Almaden
650 Harry Rd, San Jose, California 95120, United States
Email: xiaotong.liu@ibm.com

***Abstract*—Artificial Intelligence (AI) has burrowed into our lives in various aspects; however, without appropriate testing, deployed AI systems are often being criticized to fail in critical and embarrassing cases. In this paper, we propose the concept of proactive testing to dynamically generate testing data and evaluate the performance of AI systems. We further introduce Challenge.AI, a new crowd system that features the integration of crowdsourcing and machine learning techniques in the process of error generation, error validation, error categorization, and error analysis. The evaluation shows that the crowd workflow is more effective with the help of machine learning techniques.**

Keywords–Crowdsourcing; Artificial Intelligence; Proactive Testing.

I. INTRODUCTION

Artificial Intelligence (AI) becomes a technology renaissance and is beginning to solve problems in many domains. It often performs well under single-score metrics such as precision and recall. Yet, with all of the AI success, many AI applications are also criticized as they can fail in critical and embarrassing cases. For example, recent AI-powered facial recognition systems of Microsoft, IBM, and Face++ have 34% more errors with dark-skinned females than light-skinned males [1].

To address this problem, we propose *proactive testing*, a novel approach that evaluates the performance of AI models with dynamic and well-crafted dataset collected using crowd intelligence. Proactive testing differs from conventional testing metrics in two aspects. First, it extends the coverage of the testing dataset by dynamically collecting external dataset. Second, AI developers are allowed to query additional dataset belonging to certain categories to target corner cases. As a result, proactive testing is an approach to discovering unknown error and bias of a model, and providing a comprehensive evaluation of the model's performance regarding all test cases.

In this paper, we contribute a hybrid system, Challenge.AI, that combines human intelligence and machine learning techniques to assist AI developers in the process of proactive testing. Our system contains four main components including explanation-based error generation, error validation, categorization, and analysis. We bring in crowd force in error generation and encourage the crowd to craft sentences that can fail a given AI model. Especially, to assist error generation, we borrow advanced machine learning methods to explain each prediction made by the model, and present the explanation to the crowd using intuitive visualization. In addition, we employ the crowd in error validation and categorization to ensure the quality of the crafted dataset at scale. We evaluate the effectiveness of explanation-based error generation by measuring the performance of the crowd. The evaluation shows that the crowd spent less time in generating specific errors.

The structure of the rest of this paper is as follows: Section 2 discusses the related works and techniques. In Section 3 we describe a formative study to outline the challenges of model testing. Section 4 presents our Challenge.AI system, followed by an evaluation of error generation with the crowd in Section 5. We report the evaluation results of our Challenge.AI system with AI developers in Section 6, and discuss design implications in Section 7. We conclude the paper in Section 8.

II. RELATED WORK

This paper is related to prior work in three areas, e.g., generation of adversarial samples using machine learning, acquisition of corpus using crowd intelligence, and the effects of various prompts.

A. Adversarial learning for text classifiers

Several approaches have been proposed to generate adversarial examples in the deep learning community. However, most studies have focused on attacking image or audio classification models [2], [3], [4], [5], [6]. The attack of text classifiers is under-exploited due to the discrete domains involved in text [7].

To craft adversarial samples for text classifiers, some works modify the original input. For example, Liang et al. [8] proposed three perturbation strategies, e.g., insertion, deletion, and replacement to evade DNN-based text classifiers. Li et al. [9] studied the effect of removal of input text at different levels of representation. Gao et al. [10] proposed novel scoring strategies to identify critical tokens and executed a modification on those tokens. Similarly, HotFlip et al. [11] edited the input text at the character level. Ribeiro et al.[12] furthered the research by manipulating the input at the word level. That is, replacing tokens by random words of the same POS (part-of-speech) tag. Given access to the model's architecture, e.g., the computational graph, Papernot et al. [13] manipulated the output of RNN models. Although aforementioned approaches can generate sentences that fail text classifiers, the perturbation harms text integrity, resulting in unnatural and semantically meaningless text from language viewpoint.

To overcome the limitation of above methods, Samanta et al. [14] proposed a rule-based approach to ensure that the resulting text is syntactically correct. Zhao et al. [7] proposed GAN-based approach to generate adversarial input

 ISBN: 978-1-61208-785-6

that are legible to humans. The two techniques driven by machine learning are promising in their scalability. However, the resulting text has not been validated, and its quality is not guaranteed. In this paper, we design a crowdsourcing pipeline to generate and validate adversarial samples by means of human intelligence. The derived adversarial dataset is diverse from different perspectives.

B. Corpus acquisition using crowdsourcing

Online crowdsourcing provides easy and economic access to human talent [15], and has been proved effective in the acquisition of corpus in various natural language processing tasks. Some work focuses on speech transcription. For example, Parent [16] proposed a two-stage approach to transcribe large amounts of speech. Lasecki et al. [17] employ non-experts to collectively caption speech in real-time to help deaf and hard of hearing people. Others [18] proposed a variety of mechanisms to collect high-quality translations for machine translation systems, and annotate text [19], [20].

In addition, crowdsourcing has been widely applied to acquisition of paraphrasing. For example, Chklovski [21] designed a game to collect paraphrases with no prompting. Negri et al. [22] designed a set of paraphrasing jobs to maximizes the lexical divergence between an original sentence and its valid paraphrases. Buzek et al. [23] proposed the idea of error-driven paraphrasing for machine translation systems. That is, they asked crowd workers to paraphrase only the parts of the input text that are problematic to the translation system. Burrows et al. [24] focused on the acquisition of passage-level samples using crowdsourcing while Lasecki et al. [25] collected dialog dataset. Recently, Jiang et al. [26] studied the key factors in crowdsourcing paraphrase collection.

The design of Challenge.AI has been inspired by many of the above approaches. However, most previous work cannot be readily applied to acquire adversarial dataset in natural language in an iterative manner.

C. The effects of prompt

When performing a task, crowd workers are influenced by instructions, examples, and context of the task [26]. Some research focuses on how different prompts can result in natural variation of human-generated language. For example, Wang et al. [27] investigated three text-based elicitation methods, e.g., sentences, scenarios, or list-based descriptions, for collecting language that corresponds to a given semantic form. Mitchell et al. [15] explored the use of crowdsourcing to generate a corpus of natural language templates for a spoken dialog system. They investigated the effect of presenting various amount of dialog content to crowd workers. Kumaran et al. [28] explored gaming as a strategy for acquisition of paraphrase data. This work presents drawing as prompt and asks the participants to produce paraphrases. Law et al. [29] examined how crowd workers are incentivized by curiosity. In this work, we investigate how prompt can be augmented by machine learning to help crowd workers generate adversarial samples.

III. Formative Study

The goal of the formative study is to understand current practice of model testing, the challenges faced by AI developers, and potential opportunities of our system.

A. Study setup

In this study, we interviewed five AI developers (denoted as D1—D5) in an IT company who are experienced in sentiment analysis. D1 is an engineer who has built sentiment classification models for different languages, such as German, English, and French. D2 is a product manager who has analyzed errors in French sentiment models. D3, D4 and D5 are research scientists who have experience in AI model design and cross-model evaluation.

We organized semi-structured interview sessions with each expert. Each interview lasted approximately 30 minutes and covered a variety of topics, starting with a general question about their experience in sentiment analysis, followed by how they test models' performance and their observation. We also focused on the challenges they encounter and how they address them. The interviewer took notes during the interviews and recorded audios for post-interview analysis. Based on the interview results, we derived four requirements to guide the design of Challenge.AI.

B. R1: Error generation

To continuously improve the performance of sentiment models, the AI developers repeat the process of *"Build (refine) model — Train model — Test model"*, where the results of model testing guide the refinement and training of models. In model testing, the AI developers (D1, D2, D3, D4) mainly rely on metrics, such as the entire accuracy of the testing dataset, the accuracy for each sentiment category, confusion matrix, F1 score, etc. One AI developer (D3) noted, *"We built sentiment analysis models for research purpose, and evaluated our models by comparing with baseline approaches on an open dataset."* However, he was not sure about the performance of their model in real-world deployment, *"If I need to deploy our model for real use, current testing would not be enough."* Since existing testing dataset is limited in coverage, D3 suggested to borrow external dataset for a comprehensive testing, *"the intuition is that, you need to increase the diversity of the testing data so as to cover different cases."* This motivates us to employ crowd force for error generation to extend the coverage of testing dataset. In addition, we should allow AI developers to collect corpus of certain category to thoroughly test the performance of models, in particular regarding corner cases.

C. R2: Error validation

After samples are crafted by crowd force, a critical task is to decide what are their "real" sentiment and whether the model makes correct predictions. High quality testing dataset is critical for evaluating the performance of a model. Some AI developers prefer human-labeled dataset because the quality is high. That motivates us to borrow the crowd to manually validate the sentiment of each generated sample. Since the sentiment is ambiguous and subjective, we plan to employ multiple crowd workers to validate one sample and use "majority vote" to mark as the ground truth.

D. R3: Error categorization

AI developers sometimes seek to obtain samples belonging to certain category to cover corner cases. For example, D2 mentioned that, "We once tested the model for biasing. We tried Asian name and western people's names to see whether

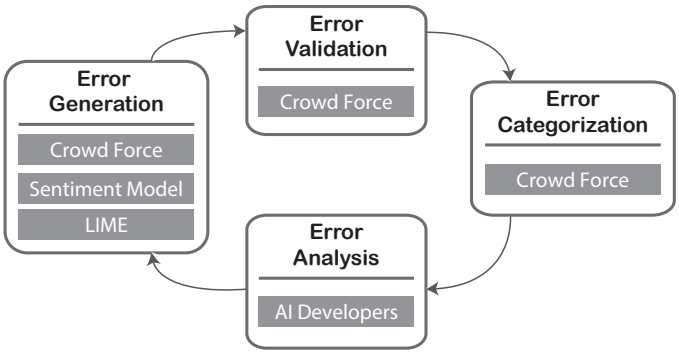

Figure 1. The architecture of Challenge.AI.

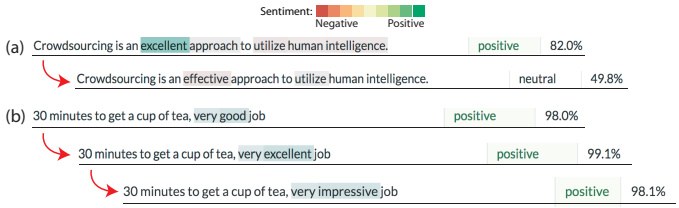

Figure 2. The usage of LIME in two cases. (a) shows how LIME helps crowd workers modify the input sentence to successfully fool the analyzer. (b) demonstrates how LIME facilitates workers to continuously generate adversarial samples.

the model would give different predictions. We also tried like female names, male name to see if there any difference." Therefore, after obtaining samples generated by the crowd, it is critical to validate the category of them. To deal with large sample size, labeling the category using the crowd is necessary to scale the labeling process.

E. R4: Error analysis

The analysis of mis-classified samples would reveal insights to the model. However, not all samples are worth the analysis. As D2 mentioned, *"If a model makes some error predictions, they may have different impact. For example, when a sentence is negative, but the prediction is positive, that would be polarity errors. People would think the model sucks. But for sentences that are ambiguous, for example, if the ground truth is positive, the model prediction is neutral, then it would be fine. Because people can understand these errors exist."* Therefore identifying mis-classified samples with high impact would help AI developers focus on the most important errors. In addition, it would be infeasible to analyze all samples due to large sample size. As a result, demonstrating the samples at multiple levels of granularities is necessary to deal with large volume of data.

IV. CHALLENGE.AI

To understand how AI developers test models in practice, we worked closely with five AI developers in an IT company who are experienced in sentiment analysis. We conducted semi-structured interviews with them to identify the challenges they face in practice. After collecting and analyzing the interview results, we concluded the following requirements: (1) Generate errors belonging to certain category. (2) Ensure the quality of the errors. (3) Categorize errors into different groups. (4) Analyze errors to reveal insights to the model. The design of Challenge.AI is guided by these requirements. Figure 1 depicts the architecture of Challenge.AI which includes four main components, i.e., explanation-based error generation, error validation, categorization, and analysis.

A. Explanation-based error generation

This component is designed to encourage the crowd to craft sentences to fail AI models for evaluating the performance of the models. When the crowd enter the error generation component, the interface shows the introduction, example sentences belonging to a certain category, and rules of this task. After reading the instruction, a worker is able to craft a sentence in the input area. The worker then presses the "Submit" button to test the performance of the model. In response, Challenge.AI launches the sentiment analysis model in the backend, and displays the sentiment label (negative, neutral, or positive) and the probability in the result panel. The worker can verify whether the model fails or not. If it fails, the worker then needs to identify the sentiment label of the sentence.

B. Accountability via machine learning

Surrounding context may have an effect in facilitating crowd workers to craft samples, affecting the performance such as efficiency, quality, and success rate [26]. We seek to augment the prompts to assist crowd workers in error generation from two aspects, i.e., starting point and accountability, respectively. The starting point refers to the existing text in the input box, we boost the crafted sentences by providing a randomly sampled error from one category. Crowd workers are encouraged to edit the sentence in the input area.

On the other hand, we provide accountability by borrowing LIME [30], an explanation technique that provides interpretable results for a prediction and is applicable to explain any models. To be specific, after a worker submits a sentence, the LIME algorithm is triggered to calculate the relationship between the prediction and each word in real time. Then the results are presented in the interface. Instead of presenting a set of numeric values, we borrow visualization techniques to intuitively depict the LIME results inline with the text. As shown in Figure 2(a), The background color of a word indicates whether it contributes to positive (green), negative (red), or neutral (yellow) sentiment.

C. Error validation and categorization

We conduct crowd-based validation and categorization by recruiting different crowd workers after the Error Generation process to obtain ground truth sentiment labels. In addition, we offer "effort-responsive" bonus to creators based on the validation results. We require at least 5 judgments for each sample, and pay $0.016 per judgment. We set up many hidden test questions for quality control, which are used to reject validations by workers who have missed a quantity of test questions [19]. The validation is performed using Figure-eight [31]

D. Error analysis

After error categorization, we obtain a dataset where each error is associated with a ground truth sentiment label validated by the crowd, a predicted label by the model, and a category.

 ISBN: 978-1-61208-785-6

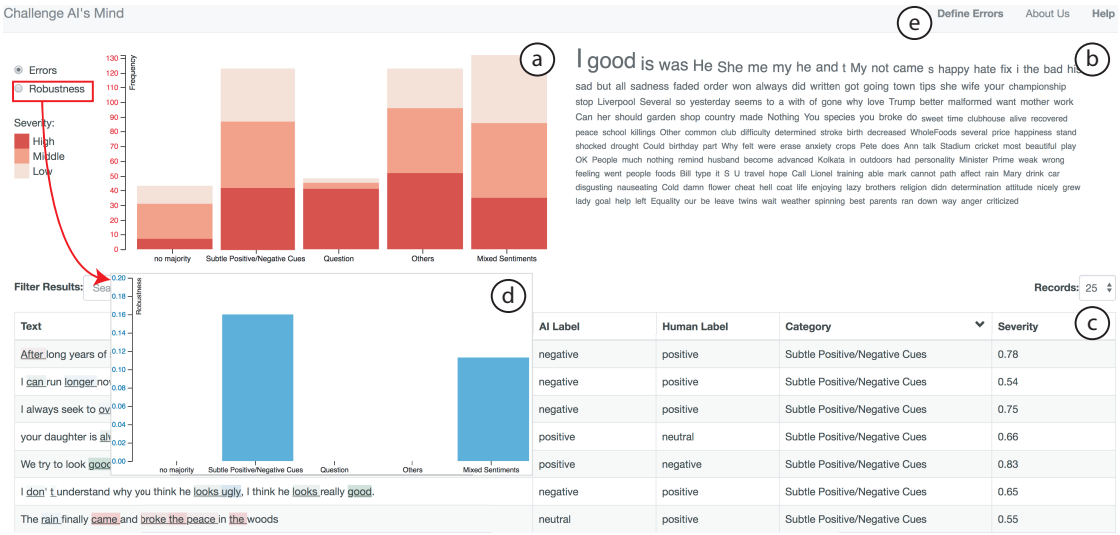

Figure 3. The interface allows AI developers to investigate the validated samples at different levels.

To understand the impact of each error, we define "Severity" for each error. The intuition is that, for a misclassified sentence, if both human and the model are confident about the sentiment, the mistake is severe. On the other hand, if both sides are not sure about the sentiment label, the mistake can be ignored. Hence, the severity score is calculated as $S = W_1 \times \text{Conf}_{human} + W_2 \times \text{Conf}_{AI}$, where W_1 and W_2 are weights for confidence of human and of the model, respectively. In this work, we set $W_1 = W_2 = \frac{1}{2}$. Conf_{human} represents the confidence of human, which is calculated as the percentage of the crowd making the judgment the same as majority vote. For example, for a sentence validated by five crowd workers, three of them validated the sentiment as positive. Hence, Conf_{human} becomes $\frac{3}{5} = 0.6$. Conf_{AI} is provided by the model, usually obtained as the probability or confidence of the prediction.

To help AI developers understand the model by analyzing a large quantity of errors, we build an interface to demonstrate the analysis at three different levels. After the data is loaded, the Statistic View (Figure 3(a)) uses a stacked bar chart to demonstrate the error distribution of each category at the macro-level. The x-axis presents different categories while the y-axis shows the number of errors. For each category, we manually set two thresholds to split errors into three classes representing different levels of severity, i.e., high (dark red), middle (light red), and low (pink). At the meso-level, a Cloud View (Figure 3(b)) shows a tag cloud summarizing sentiment words calculated by LIME [30]. The bigger a word is, the more frequent it appears in sentences as a sentiment word recognized by LIME. At the micro-level, a Table View (Figure 3(c)) demonstrates raw sentences, the prediction, sentiment ground truth, the category, and the severity. Various interaction techniques, such as linking and filtering, are borrowed to coordinate the three views.

V. Evaluation with the Crowd

We conducted a crowd evaluation to investigate how different prompts in error generation affect the performance of the crowd in crafting errors.

We constructed prompts based on different combination of accountability (LIME) and starting points (SP). If the starting point is empty, workers are encouraged to craft a sentence from scratch. Otherwise, workers are allowed to edit the text in the input area (SP).

TABLE I. STATISTICS OF ERROR GENERATION BASED ON TWO PROMPT CONDITIONS, e.g., A BASELINE CONDITION (NO LIME, NO SP) AND AN ENHANCED ONE (LIME, SP).

	LIME, SP	No LIME, No SP	Total
N_{total}	262	293	555
N_{valid}	75	108	183
#workers	66	46	112

We performed a between-subject design with two experimental conditions, e.g., a baseline condition (NO LIME, NO SP) and an enhanced prompt (LIME, SP), and identified two types of errors to generate, i.e., "Subtle sentiment cues" and "Mixed-sentiment" which refers to sentences containing both positive cues and negative indicators. We used Figure-eight [31] as the platform to release our error generation jobs. To be slightly generous, we paid \$0.05 per sentence if the sentences successfully fail the analyzer after validation. At the same time, if the sentences belong to the required category, additional \$0.05 per sentence were paid to the crowd. To reject noises and assign categories to each sample, the crowd-based validation was performed after generation.

A. Metrics

The general statistics of each job are displayed in Table I. The **Total trials**, denoted as N_{total}, include all sentences that the crowd have crafted using our system. Crowd workers have generated 249 sentences for "Subtle sentiment cues" and 306 for "Mixed-sentiment", respectively. **Validated trials** (N_{valid}) are the number of sentences that successfully fail the model based on the validation results. In addition, we count the number of distinct crowd workers for each condition (**# workers**).

 ISBN: 978-1-61208-785-6

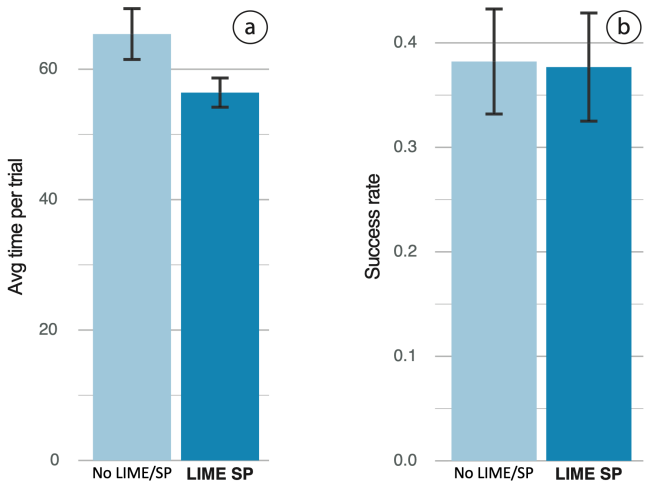

Figure 4. (a) shows the bar chart displaying average time per trial for each worker under two conditions. (b) shows how crowd workers differ in success rate. The error bars demonstrate standard errors.

Accordingly, we propose two metrics to evaluate the performance of each crowd worker. We use n instead of N to represent that the statistics values correspond to one crowd worker. **Average time per trial** (T) measures how much time that a worker needs to craft a trial on average, which indicates how efficient a sentence can be crafted. **Success rate** (R_{succ}) is measured as $\frac{n_{valid}}{n_{total}}$. This value measures how easily a worker thinks s/he can generate samples to fail the model. The success rate is useful to measure the effectiveness of prompts, as well as to analyze the vulnerability of a model.

B. Analysis of crowd performance

On average, crowd workers spent 56.4 seconds (SD=18.2) in crafting a sentence with the enhanced prompt (LIME, SP) and 65.4 seconds (SD=26.4) with (NO LIME, NO SP). Figure 4(a) shows the average time per trial (T) under each condition. We found a significant effect (t = 1.9977, p<0.05) of the enhanced prompt in reducing (T). The crowd used about 13.8% less time in crafting a sentence with (LIME, SP) than (NO LIME, NO SP). The reason may be because accountability assists workers to craft errors during the process, and editing text in the input area requires less time compared to crafting a new one from scratch. Figure 4(b) shows that crowd workers are indifferent in success rate (R_{succ}) under two conditions (38.2% V.S. 37.7%).

VI. Evaluation with AI developers

To investigate how Challenge.AI helps AI developers understand and diagnose a model, we worked with the five AI developers that we collaborated during the formative study, and organized two rounds of semi-structured interview sessions to evaluate the effectiveness and usefulness of Challenge.AI.

A. Process

We followed the architecture of Challenge.AI (Figure 1) to evaluate the entire system. Before error generation, we started from the first sessions with AI developers to obtain initial categorization for errors. Based on the category information proposed by AI developers, we used Challenge.AI to generate errors belonging to these categories, and conducted validation and categorization for crafted sentences. Finally, we organized the second interview sessions (error analysis) to understand the usefulness and limitations of Challenge.AI from the perspective of AI developers. During the entire evaluation, we used a sentiment analysis model built by D1 as the target model to test. The input of the model is a sentence, and it outputs a sentiment label associated with a probability.

1) First sessions: The goal of the first sessions is to obtain the target categories of errors to test the model. To begin with, we tested the performance of the model using a public sentiment dataset [19] where all 12284 sentences are collected from Twitter, and labeled with negative, neutral, or positive sentiment. After obtaining all misclassified sentences, we randomly sampled 200 ones and stored them in a table (CSV file format) with four columns, e.g., a 'Text' column, a 'Human_Label' column showing the ground truth, an 'AI_Label' column displaying the results calculated by the model, and an empty column titled 'Category' to allow AI developers to label a potential category for the sentence.

Each interview started with the introduction of the dataset. After that, we presented the dataset to AI developers and asked them to identify the patterns of the misclassified samples and name new categories for them. AI developers were allowed to discard sentences that are hard to be categorized. An interview took about 40 minutes. We encouraged them to express findings and thoughts using a think-aloud protocol and took notes about their feedback for further analysis.

Some AI developers have more experience in identifying patterns for errors. For example, when noticing a sentence whose benchmark label is positive, but misclassified as negative by the model, i.e., "Marissa Miller of Google makes shout out to the Khan Academy and the great things they're doing for education. #fmsignal #sxsw (cc @mention", D2 said, *"I think the model made a wrong prediction because it does not understand what 'shout out' means."* From her experience, D2 further commented that the model may not understand sentiment indications that are domain-specific or context dependent. Besides summarizing patterns in the dataset, D3 asked for sentences containing both positive and negative indicators. "Do any of them have opposite sentiment words, like, I am happy, but... something like that?" The participant further explained, *"Some models are designed to handle targeted sentiment, but determining relevant sentiment in mixed sentiment texts is challenging."* Finally we derived two categories of errors for model testing. One is called "Subtle Sentiment Cues" which means that a sentence is either positive or negative, and has positive or negative indications. The other is "Mixed-sentiment" which refers to sentences containing both positive cues and negative indicators. Further, we include three more types of errors for categorization. For example, a "Questions" category is added based on D1's comments and an "Others" is included to be more general. A "No majority" category is added after categorization if human annotators cannot reach a consensus on the category of that sample.

2) Running Challenge.AI: After obtaining the categorization, we tested the model by walking through three main components of Challenge.AI, e.g., error generation, validation, and categorization. As mentioned above, we focused on the two categories, i.e., "Subtle Sentiment Cues" and "Mixed-sentiment" in error generation while we used five categories

 ISBN: 978-1-61208-785-6

for error categorization. The results and analysis of crowd performance are described in Section V.

3) Second interview sessions: We organized second interview sessions to evaluate how Challenge.AI helps AI developers understand the performance of the model.

After running Challenge.AI, we obtained 555 samples that 112 crowd workers generated to have successfully failed the model, where 23 errors are categorized as "Subtle Sentiment Cues" and 44 are "Mixed-sentiment". During the interviews, we demonstrated the data at three levels of granularities using the interface shown in Figure 3.

Each interview took about 45 minutes. We first presented the goal of Challenge.AI to AI developers and a detailed introduction to the data and interface. AI developers then freely explored the interface and we helped them resolve any questions they encountered. Next, the participants went through the interface to tell how they understood the performance of the model. They further identified new categories of errors by investigating detailed samples using the interface. Finally, a post-interview discussion was conducted to collect their feedback about the strengths and weaknesses of Challenge.AI. During the interview, AI developers were instructed to think aloud and we took notes about their feedback. We recorded the whole interview sessions for later analysis. We report the results of second interview sessions in the remaining of the section.

B. Value of proactive testing

A thorough testing is important for AI models before deployment. However, current practice of testing is limited in coverage, as D3 commented, *"When doing the testing, we assume that the testing dataset and training dataset are in the same feature space."* Traditional testing approach is far from enough for deploying the model in the wild, which indicates the potential value of proactive testing in evaluating the model for production. To reduce critical and embarrassing errors, AI developers are able to identify corner cases to test, and Challenge.AI collect external dataset belonging to specific categories. In addition, by investigating external dataset, AI developers can discover unseen errors. For example, our participants identified two categories that are distinct from those found in the first interview session, e.g., bias in pronouns such as 'He' and 'She', and reversed sentiment containing words like 'However', 'Though', and 'But'. Detailed discussions are reported below.

C. Getting a gist

First of all, AI developers were interested in the overall patterns of misclassified samples. The Statistics View (Figure 3(a)) provides a big picture of the entire dataset. From the stacked bar chart, D5 noticed that it is about equal distribution among high severity, middle, and low for most bars. However, the samples belonging to "Question" attracted her attention because high-severity errors account for the majority in this category. *"The model could be improved (in the 'Question' category) for sure."* D5 further explained the way of improving the model, *"In some of the supervised learning models, we need to use human heuristics to do the feature engineering (extraction) from the raw dataset. The quality of the feature extracted largely impacts the final performance."* The participant took the "Question" category as an example, *"If a model a has high probability to make severe errors for question sentences, we may specify a feature in feature engineering to detect whether a sentence is a question or a statement. So with this feature, hopefully could help the model make decisions."*

From our observation of the first sessions, all AI developers had read through about a dozen of misclassified sentences because the process of error analysis requires great mental efforts. Displaying the errors at different levels of granularities would relieve AI developers in analyzing a large number of errors. As D2 commented, *"I like the overview which gives me the impression of the entire dataset. You know, reading through two hundred errors is time-consuming and impossible (during the first interview session), and I did not do a good job last time."*

D. Examining errors by words

After examining the Statistics View, D4 switched his focus to the Cloud View showing sentiment words as tag cloud (Figure 3(b)). The participant noticed that the word "I" has the biggest font size while "Good" is the second biggest word. *"Typically in sentiment analysis, you will not expect 'I' to be particularly positive or negative. 'Good' is the second one. It makes more sense but 'I', 'is', 'was', 'he', 'me', 'my', 'she', among the first line are not sentiment words."* However, the participant changed his mind after investigating sentences containing "He" and "She". He first clicked "She" and the Table View updated. The participant noticed that the word contributes a lot to neutral sentences, and contributes once for negative and positive, respectively. Similarly, the participant further examined sentences containing the word "He", and noticed that four out of eight are negative, and "He" contributes to the negative sentiment. *"Well, it is interesting to see the difference between 'She' and 'He'. I guess the model tends to regard 'He' as a negative word."* He added, *"I think that it is necessary to examine the training data (of the model) to see whether the stop words are equal in distribution for each sentiment."*

Before using Challenge.AI, some AI developers (D1, D4, and D5) found it hard to identify patterns and categorize sentences. For example, during the first interview sessions, D4 did not know the reason for some of the predictions. The participant pointed to one question sentence and commented, *"There is no reason to label this question into negative or positive. Because it apparently contains none of the words with any sentiment."* D4 and D5 noted that they did not agree with some ground truth labels. As D4 said, *"I would recommend you have a category for mis-labeled because it is subjective."* The participant further pointed to a sentence whose benchmark label is neutral, and added, *"Now here is one, 'Social Is Too Important For Google To Screw Up A Big Launch Circus'. It sounds kind of negative to me, which is how the model classified it as."* By borrowing LIME [30] to extract sentiment words, Challenge.AI provides explanation of errors at the word level, allowing AI developers to find potential bias in the training data.

E. Reading through errors

D1 showed great interest in the exploration of samples in the "Mixed-sentiment" category. He clicked bars with dark red color under this category and read through these severe errors in the Table View. Then the participant noted, *"Some*

sentences in this category are reversed sentiment." Then the participant pointed to a sample and added, *"Like in this case, it has the word 'but'. All content after 'but' is the content that the speaker wants to emphasize. The former part is like warm up. So the later part highlights the whole meaning of the sentence. In this case, I will not say it is a mixed sentiment. It is reversed."* Then, the participant used the search box to find all sentences containing "however" but found no sample in the table. He commented, *"I would like to test the model with sentences using reversing words, like 'but', 'however', 'although'. The model may not do a good job."*

During the first interview sessions, we realize that not all errors are worth investigation. When looking at the errors, D5 commented, *"A lot of these are difficult for human. For those which are less obvious, you may ask three different people and got three difficult answers."* The participant further added, *"Since sentiment analysis is subjective, if an error is ambiguous to human, I do not think the model made a severe mistake."* Therefore, the definition of severity helps AI developers focus on errors that are important to examine.

VII. Design Implications

Proactive testing is a promising direction that helps AI developers get more insights into the model. Challenge.AI is the first prototype that supports proactive testing using the crowd force, and we suggest the following aspects that future research can explore.

First, *include all the generated data by the crowd including those that can fail the model and those cannot.* Because only the misclassified samples are not enough to help AI developers understand how the model performs in some cases. For example, D2 has found two sentences containing the word "Trump" by filtering. However, the participant could not conclude whether the model is biased to the word "Trump". D2 commented, *"I am only looking at the errors. It is hard to tell (whether the model is biased to "Trump"). I mean, these errors could be 99% of the instances in which case the model is doing very poorly. But this could be less than 1% of the instances in which case the model is doing fantastic."*

Second, *apply better explanation techniques.* In this study, we choose the LIME algorithm [30] to identify and highlight sentiment words related to the prediction. However, our participants found that some sentiment words are confusing. For example, D4 found a positive sentence with AI labeled negative, "I can run longer now". The word "can" is highlighted in green (positive) and "longer" highlighted in blue (neutral). He commented, *"The AI label is negative. However, it is wired that no words are marked as negative."* However, when more advanced analytical techniques are developed in the future, such issue may be resolved.

Third, *enhance the generation component for word-level categories.* Challenge.AI has been proved to be effective in collecting samples belonging to *concept-level* categories such as "mixed-sentiment" and "subtle sentiment cues". However, AI developers may sometimes seek to test the model using samples containing certain words, such as "Trump". Intuitively, collecting samples with certain words could be more cost- and time-efficient by using techniques in information retrieval. We plan to study how various information retrieval techniques help in collecting samples of different category.

Fourth, *provide real-time feedback for proactive testing.* The main process of sample collection, e.g., generation, validation, and categorization, takes a long time and AI developers cannot test the model in real-time. One possible solution is to borrow workflows from real-time crowdsourcing [32], [33], [34], [35] to reduce the delay in obtaining the testing results. Another solution is to augment the error analysis interface as suggested by D2, *"Since the model is already trained. Maybe you can (embed the model in the backend and) add an input box for real-time testing so that I can test some of the sentences in my mind."*

Fifth, *augment error analysis with advanced analytical methods.* our system borrows knowledge from AI developers to identify new patterns to test. However, the process is time-consuming and not scalable. It would be beneficial to incorporate automatic analytical methods, such as text classification or clustering, to assist AI developers in summarizing patterns among errors.

VIII. Conclusion and Future Work

To summarize, we propose Challenge.AI, a crowd system that supports proactive testing for AI models by extending the coverage of testing dataset with crowd-generated errors. To assist error generation, we propose an explanation-based error generation technique combining human intelligence and machine learning. We use crowd evaluation to compare the explanation-based error generation technique and a baseline approach. In the future, we plan to establish metrics to compare the generated dataset and open sourced ones from different perspectives, such as the topic coverage, syntactic structure, and uni-gram distribution, to have a comprehensive understanding of the crowd-crafted dataset.

References

[1] https://www.forbes.com/sites/parmyolson/2018/02/26/artificial-intelligence-ai-bias-google/#5e6155b31a01, retrieved: 2020-02-20.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[3] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and music adversaries," IEEE Transactions on Multimedia, vol. 17, no. 11, 2015, pp. 2059–2071.

[4] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," arXiv preprint arXiv:1607.02533, 2016.

[5] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," arXiv preprint arXiv:1605.07277, 2016.

[6] N. Papernot and et al., "Practical black-box attacks against machine learning," in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 2017, pp. 506–519.

[7] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," arXiv preprint arXiv:1710.11342, 2017.

[8] B. Liang and et al., "Deep text classification can be fooled," arXiv preprint arXiv:1704.08006, 2017.

[9] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," arXiv preprint arXiv:1612.08220, 2016.

[10] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," arXiv preprint arXiv:1801.04354, 2018.

[11] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, 2018, pp. 31–36.

 ISBN: 978-1-61208-785-6

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in AAAI Conference on Artificial Intelligence, 2018.

[13] N. Papernot, P. McDaniel, A. Swami, and R. Harang, “Crafting adversarial input sequences for recurrent neural networks,” in Military Communications Conference, MILCOM 2016-2016 IEEE. IEEE, 2016, pp. 49–54.

[14] S. Samanta and S. Mehta, “Towards crafting text adversarial samples,” arXiv preprint arXiv:1707.02812, 2017.

[15] M. Mitchell, D. Bohus, and E. Kamar, “Crowdsourcing language generation templates for dialogue systems,” Proceedings of the INLG and SIGDIAL 2014 Joint Session, 2014, pp. 172–180.

[16] G. Parent and M. Eskenazi, “Toward better crowdsourced transcription: Transcription of a year of the let’s go bus information system data,” in 2010 IEEE Spoken Language Technology Workshop, 2010, pp. 312–317.

[17] W. Lasecki and et al., “Real-time captioning by groups of non-experts,” in Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, ser. UIST ’12. New York, NY, USA: ACM, 2012, pp. 23–34. [Online]. Available: http://doi.acm.org/10.1145/2380116.2380122

[18] O. F. Zaidan and C. Callison-Burch, “Crowdsourcing translation: Professional quality from non-professionals,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, ser. HLT ’11. Association for Computational Linguistics, 2011, pp. 1220–1229. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002626

[19] S. Rosenthal, N. Farra, and P. Nakov, “Semeval-2017 task 4: Sentiment analysis in twitter,” in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 502–518.

[20] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, “Semeval-2016 task 4: Sentiment analysis in twitter,” in Proceedings of the 10th international workshop on semantic evaluation (semeval-2016), 2016, pp. 1–18.

[21] T. Chklovski, “Collecting paraphrase corpora from volunteer contributors,” in Proceedings of the 3rd International Conference on Knowledge Capture, ser. K-CAP ’05. ACM, 2005, pp. 115–120. [Online]. Available: http://doi.acm.org/10.1145/1088622.1088644

[22] M. Negri, Y. Mehdad, A. Marchetti, D. Giampiccolo, and L. Bentivogli, “Chinese whispers: Cooperative paraphrase acquisition.” in LREC, 2012, pp. 2659–2665.

[23] O. Buzek, P. Resnik, and B. B. Bederson, “Error driven paraphrase annotation using mechanical turk,” in Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk. Association for Computational Linguistics, 2010, pp. 217–221.

[24] S. Burrows, M. Potthast, and B. Stein, “Paraphrase acquisition via crowdsourcing and machine learning,” ACM Trans. Intell. Syst. Technol., vol. 4, no. 3, 2013, pp. 43:1–43:21. [Online]. Available: http://doi.acm.org/10.1145/2483669.2483676

[25] W. S. Lasecki, E. Kamar, and D. Bohus, “Conversations in the crowd: Collecting data for task-oriented dialog learning,” in First AAAI Conference on Human Computation and Crowdsourcing, 2013.

[26] Y. Jiang, J. K. Kummerfeld, and W. S. Lasecki, “Understanding task design trade-offs in crowdsourced paraphrase collection,” arXiv preprint arXiv:1704.05753, 2017.

[27] W. Y. Wang, D. Bohus, E. Kamar, and E. Horvitz, “Crowdsourcing the acquisition of natural language corpora: Methods and observations,” in 2012 IEEE Spoken Language Technology Workshop (SLT), 2012, pp. 73–78.

[28] A. Kumaran, M. Densmore, and S. Kumar, “Online gaming for crowdsourcing phrase-equivalents,” in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 1238–1247.

[29] E. Law, M. Yin, J. Goh, K. Chen, M. A. Terry, and K. Z. Gajos, “Curiosity killed the cat, but makes crowdwork better,” in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ser. CHI ’16. New York, NY, USA: ACM, 2016, pp. 4098–4110. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858144

[30] M. T. Ribeiro, S. Singh, and C. Guestrin, “"why should I trust you?": Explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, 2016, pp. 1135–1144.

[31] https://www.figure-eight.com/, retrieved: 2020-02-20.

[32] W. S. Lasecki, K. I. Murray, S. White, R. C. Miller, and J. P. Bigham, “Real-time crowd control of existing interfaces,” in Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, ser. UIST ’11. ACM, 2011, pp. 23–32. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047200

[33] G. V. de la Cruz, B. Peng, W. S. Lasecki, and M. E. Taylor, “Towards integrating real-time crowd advice with reinforcement learning,” in Proceedings of the 20th International Conference on Intelligent User Interfaces Companion, ser. IUI Companion ’15. New York, NY, USA: ACM, 2015, pp. 17–20. [Online]. Available: http://doi.acm.org/10.1145/2732158.2732180

[34] A. Lundgard, Y. Yang, M. L. Foster, and W. S. Lasecki, “Bolt: Instantaneous crowdsourcing via just-in-time training,” in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ser. CHI ’18. New York, NY, USA: ACM, 2018, pp. 467:1–467:7. [Online]. Available: http://doi.acm.org/10.1145/3173574.3174041

[35] Z. Liao, Y. Xian, X. Yang, Q. Zhao, C. Zhang, and J. Li, “Tscset: A crowdsourced time-sync comment dataset for exploration of user experience improvement,” in 23rd International Conference on Intelligent User Interfaces, ser. IUI ’18. New York, NY, USA: ACM, 2018, pp. 641–652. [Online]. Available: http://doi.acm.org/10.1145/3172944.3172966

 ISBN: 978-1-61208-785-6