First Impressions Matter: A Voice Assistant Study Using Log-Based Predictive Metrics

Tapas Kanungo, Nehal Bengre, Stephen Walsh,
Qingxiaoyang Zhu and Dmitrii Siakov

Abstract

Voice assistants like Siri, Alexa, Google Assistant, and Bixby are increasingly becoming dependable services for accomplishing simple tasks. However, the quality of these systems is still far from where we would like it to be. Thus it is important to monitor user engagement and find ways to identify user experience issues using online user logs. In this paper, we present a study using log-based predictive metrics that shows that a user's first interaction quality can dramatically impact the user retention rate. We use the predictive metric – task completion rate in our study — to identify the Bixby Apps that are not performing well in real-time. The task completion rate itself is estimated using a machine-learned system that is built using a small sample of user logs with corresponding quality labels annotated by human graders. Furthermore, we also show how to use the user task completion rates to monitor the user base satisfaction rates over long periods. Finally, an additional insight from our finding is that the first few days of user interaction can give us a very good source of information to mine for causes of user dissatisfaction and retention loss.

Tapas Kanungo
Samsung Research America / Mountain View, CA, USA e-mail: tapas.k@samsung.com

Nehal Bengre
Samsung Research America / Mountain View, CA, USA e-mail: n.bengre@samsung.com

Stephen Walsh
Samsung Research America / Mountain View, CA, USA e-mail: s1.walsh@samsung.com

Qingxiaoyang Zhu
University of California Davis / Davis, CA, USA, e-mail: qinzhu@ucdavis.edu

Dmitrii Siakov
Samsung Research America / Mountain View, CA, USA e-mail: d.siakov@samsung.com

1 Introduction

Online voice assistant services like Siri [1], Alexa [2], Bixby [3] and Google Home [4] are ubiquitous today. The predominant use of such assistants is executing simple tasks like switching on the lights, calling someone, or setting an alarm. Around 70% of Bixby's sessions are single turns. It is thus important for all assistants to understand if their users are completing their tasks or not and how their task completion rate relates to user retention. In particular, in this work, we wanted to know if users were negatively impacted if their initial interaction with Bixby was poor.

While offline evaluation of user satisfaction or task completion using human-graded labels is commonplace, the process is slow, laborious, and prohibitively expensive. Thus it is not possible to have a large corpus of online user logs that have human-quality labels. Any quality-based log analysis would need to be done using a predictive metric that uses a machine-learned model to predict the user satisfaction rate using online features.

To achieve this goal we built a machine-learned model to predict the task completion rate from the user logs in real-time instead of humans grading the logs at periodic intervals. Typical "conversations" with Bixby are single turns and many Bixby responses are non-verbal, e.g. a beep or flash to indicate the task was completed or internal state variables. We explored transformer-based models and hybrid models using BERT embeddings to predict task completion rate. In addition to the BERT embeddings of the user utterances and system responses, we use contextual system features and non-verbal cues as features and train a Random Forest to predict the grader labels.

To apply this model to the logs in real time, we compute the scores on each conversation turn in the log and declare a conversation task to be completed satisfactorily (SAT) if the score is in the top quartile, and the task to be completed unsatisfactorily (UnSAT) if the score is in the bottom quartile.

To understand the relationship between the initial task completion rate and the user retention rate, we analyzed six months of Bixby user logs and created cohorts of users by aligning their start date, which is the date they are first observed in the logs and have at least one month of no activity. Next, we identify the user's most used (popular) app in the first day and first week of using Bixby. We then create two cohorts — users whose highest score was in the bottom quartile (UnSAT) and those whose highest score was in the top quartile (SAT) — and compute these cohorts' retention rate as a function of time. As one would have predicted, we find that users in the SAT cohort have a higher retention rate. Next, we analyze the time-to-return after the first encounter for the two cohorts. We again find those SAT users are more likely to return faster than UnSAT users. We show that SAT and UnSAT rates can be used for monitoring the health of the user base over long periods.

[1] https://www.apple.com/siri/

[2] https://alexa.amazon.com/

[3] https://www.samsung.com/us/apps/bixby/

[4] https://assistant.google.com/

The contributions of our paper are summarized as follows:

- A predictive machine-learned task completion metric that estimates user satisfaction in near-realtime from online user logs instead of using human graders.
- An in-depth analysis and study of user retention over six months of Bixby online user logs as a function of predicted task completion rate. It is demonstrated that the predictive metric is a viable way to monitor aspects of user experience quality over time.
- Insight that the first few days of user enrolment can give valuable information about user dissatisfaction and low user retention rates.

In the next section, we describe the related literature in Section 2. Next, we give an overview of the Bixby system and environment in Section 3. We then follow with details of task completion modeling in Section 4. We then present a time-based analysis of Bixby in Section 5. Finally, we characterize user retention as a function of task completion rate in Section 6.

2 Related Literature

The main contribution of this paper is a study of online user engagement over time. To conduct this study we partition users into cohorts by nature of the initial quality of user engagement. In this section, we will first review related work in the online metrics and retention area. Next, we will describe related literature on predictive metrics for voice assistants.

In the image classification area, user trust was negatively impacted when they encountered errors early in the interaction [24]. Similarly, in robotics, it has been found that the initial unreliability of the robot can negatively impact the trust the user has on the robot [5]. Both these papers have findings similar to ours but in different settings other than voice assistants. Since the user interaction models in voice-assistants are different, it is not clear that the result from robotics and image classification will easily generalize to voice assistants. In [3] authors present a model to predict user return probability as a function of whether the search engine provided the user with a long duration click in the search session — their heuristic for a satisfied user. The finding was that a user with a long-duration click has a much higher return probability than a user with no long-duration click. In [11] the authors present a methodology to predict user retention in a web search scenario and compare two cohorts – stationary and non-stationary. Both papers study web search user scenarios, not voice assistant task completion scenarios. In addition, they do not analyze based on the quality of the first use. Web search queries typically have multiple page views and so have a different engagement model compared to voice assistants in task completion scenarios. Finally, while the field of A/B testing [17, 18, 8] uses online metrics to compare control and treatment populations chosen randomly, our work studies the impact of initial engagement quality on users over time.

While we use task completion as our predictive metric, any other user satisfaction metric can be used. For completeness, we now review papers that have presented research on user satisfaction prediction but have not used it for user retention analysis. In this work, we did not consider open-ended chat-oriented dialogue Apps as it typically has multiple turns, and as stated in Section 1, Bixby system user logs has a majority of the single- or few-turn dialogues. Online satisfaction metrics were presented in [16, 15, 10] where the authors designed a user study and asked users to perform certain tasks using the Cortana voice assistant. The logs of these user interactions were then used to build a predictive model for user satisfaction. In practice, however, the user queries and interaction behavior in artificial setups are quite different from what one sees in real-world systems. For example, the length and nature of the questions are quite different in the two setups. Our approach, in contrast, trains a model using graded labels for real user logs and thus is more representative of the real-world engagement model. In addition, our work shows how task completion scores can predict user retention. A related work [22] also proposes an online user satisfaction model for voice assistants and [29] proposed a holistic metric to evaluate the performance of dialogue agents taking task success and dialogue costs into account, however, they do not show how it can be used for estimating user retention. Work on task completion [30] while related does not present an online predictive metric or study its behavior concerning user retention. Similarly, task completion work presented in [20, 25, 19] presents predictive scoring techniques, but does not use them for conducting any longitudinal study and does not present any insight regarding long-term user retention.

3 Bixby Platform Overview

The Bixby[5] voice assistant service has around 100 "basic" Apps [6] and over 1000 third-party Apps. Some of the most popular Apps and corresponding sample utterances are: Phone ("call mom"), Clock ("set an alarm for 6 a.m."), System ("turn up the volume"), Launcher ("open Netflix") and QnA ("when was Barack Obama born"). While Bixby also has a Chat App, we did not include it in our study since the Chat App interactions are typically open-ended conversations and do not have a specific task that needs to be completed. Bixby's online service is available on multiple consumer devices (e.g. phones, TV, watch, fridge, PC, etc.) and in multiple languages.

A high-level architecture of the Bixby voice assistant is shown in Figure 1. The sequence of events that happen after a user invokes Bixby are as follows. The client service first authenticates the user. Next, the speech data is sent for automatic speech recognition (ASR). The textual response is then sent to Bixby Operating System (BOS), which first classifies the utterance to the appropriate App, and then identifies

[5] The Bixby Studio is accessible via https://bixbydevelopers.com/

[6] a.k.a. Skills and Actions.

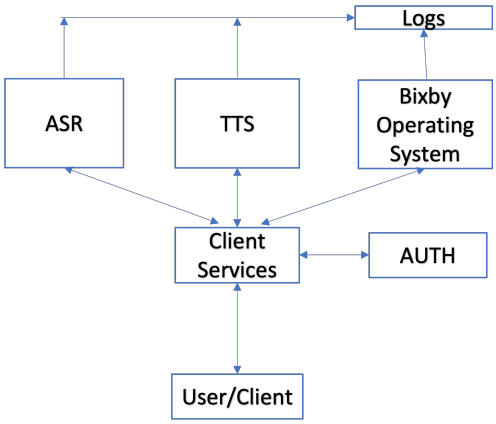

Fig. 1: Bixby system overview.

the appropriate intent and slots. BOS also maintains the state of the dialogue. The parsed utterance is next sent to the planner, which creates an execution plan for Bixby to execute (e.g. for the utterance "switch on the light" Bixby will figure out what needs to be done to physically switch on the light).

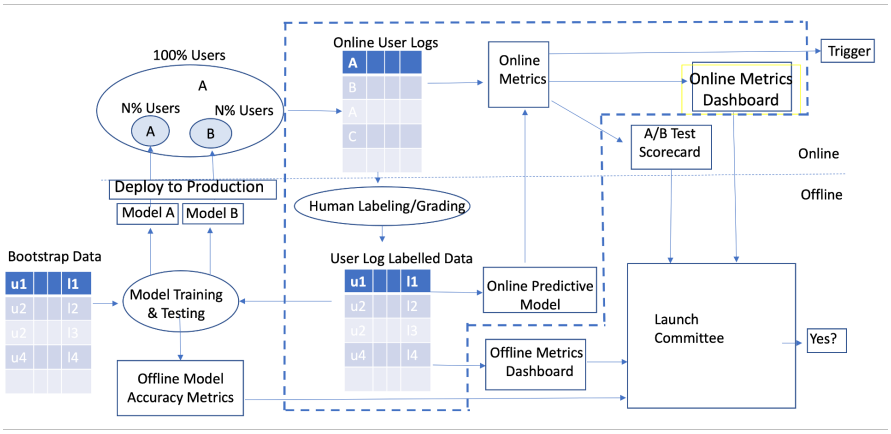

Fig. 2: Bixby service overview.

It is important for any system architect to have the appropriate instrumentation for diagnosing and identifying root causes for low user retention. In Figure 2 we show an overview of the Bixby service architecture that logs user feedback for our retention and monitoring study. The offline part is everything that is not in real-time. The online part happens in real time. This paper deals with only the section that is

inside the dashed lines. While the A/B testing component is still in development, the online metrics system is the system that analyzes for understanding user issues.

In the next section, we describe our task completion model that we use to predict task completion rate in real-time, and in Section 6 we use it to conduct a longitudinal study of user retention as a function of the initial task completion rate.

4 Task Completion

In this section, we first define task completion and provide human grading guidelines that graders used to label each user utterance. The binary task completion labels represent whether or not the human grader thinks that the Bixby user completed their task. In the following subsections, we describe the methodology we used to build machine-learned classifiers that generate a score which is an estimate of the probability of user task completion. In subsection 4.2 we describe the textual and non-textual features we used for building our classifier. In subsection 4.3.1 and subsection 4.3.2 we describe the details of the two deep-learned modeling techniques and in subsection 4.4 we describe our reasoning for picking the final model.

4.1 Data Collection

Data was collected and sampled from Bixby's user logs. The process involved uniformly sampling conversations from the logs per hour. A team of internal graders who are linguists annotated 45.2k user utterances (with Bixby's natural language understanding result – domain, intention, slot) for satisfactory or unsatisfactory task completion based on whether or not:

- Bixby's response fulfilled the user request, and
- Bixby identified the correct App, intent, context, and correct slots.

Each utterance was graded by two graders from a team of fourteen linguists using the above guideline and any conflicts were resolved by a senior reviewer. One of two labels (SAT, for satisfactory task completion, and unSAT for unsatisfactory task completion) was assigned to each utterance by the graders. We used a set of 17768 utterances for which we had grader ids logged for estimating the inter-rater agreement. Since we have multiple raters, Cohen's kappa [4] is not appropriate. Furthermore, since only two graders out of the fourteen grade each item, there are 12 missing entries per item, which rules out the use of Fleiss' kappa [7]. We used Gwet's agreement coefficient estimate AC1 [9] instead since it allows for multiple graders and missing values. The AC1 inter-rater agreement prior to the senior rater resolving conflicts was 0.94, which reflects a high inter-rater agreement.

The dataset was skewed towards the SAT cases and had a proportion of 4.3:1 with UnSAT cases. We initially performed a 9:1 dataset split into train and test. We

then downsampled the training set SAT cases to match UnSAT cases. We created a 9:1 split of the training set to create a holdout set and upsampled the test set's UnSAT cases to match its SAT cases.

Thus, the machine-learned model's objective is to produce a score in the interval $[0, 1]$ that represents the probability that the true label is a 1 (SAT, or task was completed satisfactorily). The details of the modeling process are described in the following subsections.

4.2 Features

When Bixby collaborates with a user to finish a task, the factors influencing the task completion rate are not only how Bixby responds to the user's request, but also the type of the request and the interaction patterns between the user and Bixby. These factors can be text utterances and categorical or numerical features. Therefore, we created textual and non-textual features that can be used in predictive models to evaluate the task completion rate. We leverage the textual representations of the user utterance and Bixby's response and designed the following *textual features*:

- BERT language model embeddings for user and Bixby utterances.
- Similarity of the user utterance to the previous user utterance. This shows that the user repeated their request and indicates that Bixby did not provide a satisfactory response to the previous user request
- Sentiment of the user utterance [12][7]
- We use the following Bixby Dialogue Act states as features for general multi-turn scenarios in the system:
  - Order: The user asks the Bixby to perform some task (e.g. calling, booking a ride, etc).
  - Choice: The user selects a choice from a list of options provided by the Bixby.
  - Affirmation: The user accepts Bixby's request.
  - Negation: The user declines Bixby's request.
  - Reply: User responds to Bixby's question.
- Dialogue act of Bixby. We create a representation for identifying the intent of Bixby's response into a set of Dialogue Acts:
  - Result: Bixby responds to the user query with a result.
  - Confirmation: Bixby asks the user to confirm its future action.
  - ActionFailed: Bixby fails to execute an action,
  - Selection: Bixby asks the user to select from the provided options.

In addition, we used the following *numeric features* as a proxy for users' interaction patterns:

[7] The sentiment package of NLTK is used to detect the sentiment intensity of the utterance [12].

- Day of the week
- Hour of the day when users issued the request
- Ratio of words to the character of the user utterance
- Number of stop-words [8]
- If the user utterance was a Wh question or not
- Which App (eg: Clock, Phone, Q&A, etc)

Besides the above numeric features representing the interaction patterns, we also use the following categorical features that are logged by Bixby and are thus available for real-time inference:

- App name (eg: Clock, Phone, Q&A, etc)
- User utterance and previous user utterance
- Bixby utterance
- Bixby Dialogue Act states
- User-request timing details

4.3 Modeling Methods and Performance Results

To create a predictive metric for task completion, we built a binary classifier (to classify the utterance as SAT or unSAT representing satisfactory or unsatisfactory task completion) with the aforementioned textual/non-textual features using labeled data. Thanks to the evolution of large language models [27, 23, 26, 21, 6], large-scale pre-trained language models provide a better representation of the sentence's semantic meaning and the ability to extract contextualized embedding to featurize the user utterance. In particular, we leverage the strength of BERT [6], one such large language model, to extract text representation from the user utterance. However, since the large language models are trained on text corpus only, they do not have any prior representation of our specific numerical features, making the joint representation of textual and numeric features challenging.

To jointly represent the heterogeneous features, we investigated two strategies for building a task completion prediction model:

- **Hybrid Random Forest**: Use Random Forest [2] as the backbone of the classifier. Extract contextualized sentence embedding of the text using BERT[9], then feed text representation vectors and numerical and categorical features[10] into the hybrid Random Forest as inputs. Then use the task completion label to train the forest;

[8] Stop-words from corpus package of NLTK is used to detect the stop-words in the utterance.

[9] We pick up the last layer hidden state of `[CLS]` token to represent the sentence [1]

[10] Some numerical features are derived based on the extracted text vector, such as similarity

- **Fine-Tuned BERT**: Use BERT as the backbone of the classifier. Add a classification layer on top of the original BERT[11], use the original text and numerical features as input, and fine-tune the model on the task completion dataset.

We provide the modeling details in the following subsections.

4.3.1 Hybrid Random Forest

Random Forest is an ensemble algorithm that builds multiple decision trees and gets a majority vote based on the trees' predictions. Since the decision trees in a random forest can be trained in parallel, training a random forest is generally fast and efficient. In a production setup like ours, this is a good model candidate both in terms of latency as well as performance.

After extraction of the textual and non-textual features, we concatenate the textual and non-textual features into a single 1154 (768 BERT embedding + 386 feature size) dimension vector and train a Random-Forest-based classifier. We perform a hyper-parameter grid search over the following random forest parameters: number of trees, the minimum number of samples for the leaf node, the minimum number of samples to split an internal node and maximum depth. The BERT-based model gives sufficient prediction results on task completion as seen in Table 1.

Table 1: Test set metrics for different modeling methods

Models	Class-Label	Precision	Recall	F1-score
Hybrid Random Forest	SAT	83%	98%	90%
	unSAT	97%	80%	88%
Fine-tuned BERT	SAT	89.2%	94.4%	91.7%
	unSAT	94%	88.6%	91.2%

4.3.2 Fine-tuned BERT

Our BERT model accepts multiple sentences and context-relevant features, such as domain and user interaction time. In addition, our model allows additional numerical feature. To connect multiple textual segments into BERT, for example, user request and system response, we followed the following strategy: Add a pre-designed prefix template to each text segment so that the transition between each segment is smoother. For example, prepend a role label "The user said:" to the user's utterance. Then, use (`[SEP]`) to connect the full segments.

Based on the aforementioned features, we experimented with various combinations to build the classifier with the (user utterance, system response) pair. Then,

[11] We use uncased base BERT for this study

we added the previous user utterance and task domain. According to the feature selection result, we find that the system with current and previous user turns along with the system turn utterances performed the best. The hyper-parameter is further tuned on batch size and learning rate. The best model is trained with AdamW optimizer with $1.05\mathrm{e}{-5}$ learning rate and 16 batch size. Model fine-tuning converges in one epoch and performs well in predicting the task completion label on the test data split as seen in Table 1. Compared to the hybrid Random Forest, which was trained using the features mentioned in subsection 4.2, the fine-tuned BERT model performs well even without explicitly including similarity, sentiment, and task domain features. The possible explanation could be that BERT can learn these feature implicitly from the inputs.

Table 2: Test set accuracy for different modeling methods for single-turn and multi-turn conversations

Type of model	Single-turn accuracy	Multi-turn accuracy
Hybrid Random Forest with hand-crafted features	97.61%	80.73%
Fine-tuned BERT with current and previous user turns with system turn	94.33%	88.55%

4.4 Analysis and Model Selection

Next, we compared the task completion prediction scores from hybrid Random Forest and fine-tuned BERT models on single-turn and multi-turn conversations. While most of the conversations (67%) in the test set are single-turn conversations, only 17% of the conversations are two-turn. As seen in Table 2, fine-tuned BERT performs better than hybrid Random Forest in multi-turn conversations. This difference in performance can be attributed to incorporating previous user utterances as a feature in the BERT fine-tuning process. However, hybrid Random Forest performed better than fine-tuned BERT in single-turn conversations. Since around 67% of Bixby dialogues are single-turn, we used the hybrid Random Forrest score to estimate the task completion probability [12].

To implement the task completion prediction in real-time using user logs, we had to optimize the latency of the hybrid Random Forest classifier. The most time-consuming part of the system is the BERT sentence embedding extraction, which is described in Appendix 3.

[12] We are currently working on improving the fine-tuned BERT model on single-turn utterances.

5 Session Analysis

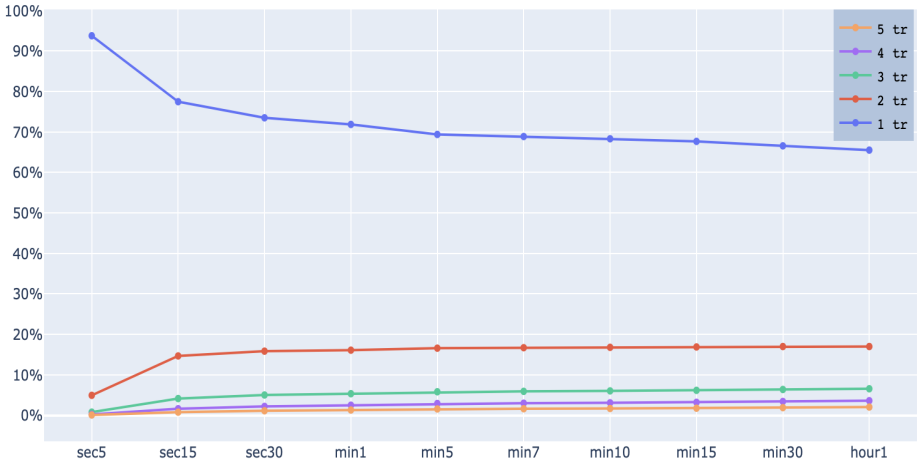

Fig. 3: Session analysis. The x-axis indicates the time threshold between two consecutive user utterances. "tr" in the legend indicates turn (interaction between the user and system).

To characterize Bixby user engagement, we first looked into the multi-turn pattern. That is, how many times the user says only one utterance, how many times the conversation progresses to two turns, etc. The difficulty of doing this experiment is in defining what is a session. One-turn session means a single interaction between a user and the system.

We used a time threshold between two consecutive user utterances to demarcate sessions. Depending on the time threshold used, we get different turn distributions. This distribution profile is shown in Figure 3. As can be seen, if we use a 30-minute threshold to separate sessions (no user activity for 30 minutes), around 67% of all sessions are just one-turn sessions, and around 18% of all sessions are two-turn sessions, etc. Thus significant number of users are currently not engaging in conversations but probably issuing one-utterance tasks or getting frustrated by the system response.

To investigate further, we wanted to understand the initial interaction of the user better. In particular, we wanted to quantify the users' initial task completion satisfaction, and whether it had an impact on user retention. One way to quantify this is to come up with a predictive metric that estimates the task completion score and see how that impacts user retention, which is described in the next section.

6 User Retention

In this section, we describe three analyses that help us understand user retention better. In the first analysis, we first aligned all users by their start date. Then we created two cohorts (as discussed in the following paragraph) based on their experience in an initial period of 1, 7, or 21 days and the most popular App the user used in that initial period. In our analysis users are considered 'new users' if they have 28 consecutive days of inactivity.

The first cohort is composed of all users who received a maximum task completion score[13] that was in the top quartile of the overall score distribution in the initial time window. Users in this cohort most likely had a good overall initial experience. The second cohort was composed of users whose maximum task completion score was in the bottom quartile. Users in this cohort most likely did not have a good initial experience. Next for each cohort, we computed the user retention plot where each point represents the probability of a user returning between that day and end of the experiment date. Figure 4 shows the behavior of retention over popular Apps. Each labeled chart in the figure represents the aggregation of all users who had that App as the most popular one over the initial scoring period, and the Apps are sorted based on popularity. The y-axis is the percentage of users who return at or after the number of days shown on the x-axis. We can see that i) the task completion experience in the initial period is a good predictor of future retention. If the users do not have a good experience they tend to exit the system early.

We then wondered what impact the initial experience has on the time to return. We used the same cohorts as above and studied how many users return as a function of time. In Figure 5 we show the percentage of users who return to interact with the system after the initial interaction. The blue curve represents users in the top quartile, while the red curve is the bottom quartile. This chart shows the percentage of users who return (y-axis) on or before the number of days shown on the x-axis. We see that initial impression matters. If the users have a good experience, they tend to return faster to re-use the system.

Finally, in Figure 6 we show i) the relative percentage (with respect to the SAT percentage in the first interaction) of users who receive a maximum score in the top quartile, and ii) the relative percentage of users who receive a maximum score in the bottom quartile. The y-axis is the relative change of the SAT score in both the top and bottom quartiles from the initial date at the end of August. We see that overall the relative SAT percentage is increasing and the relative UnSAT percentage is decreasing which indicates that overall user satisfaction is improving.

[13] We predict the task completion score for each user utterance in a dialogue and take the maximum.

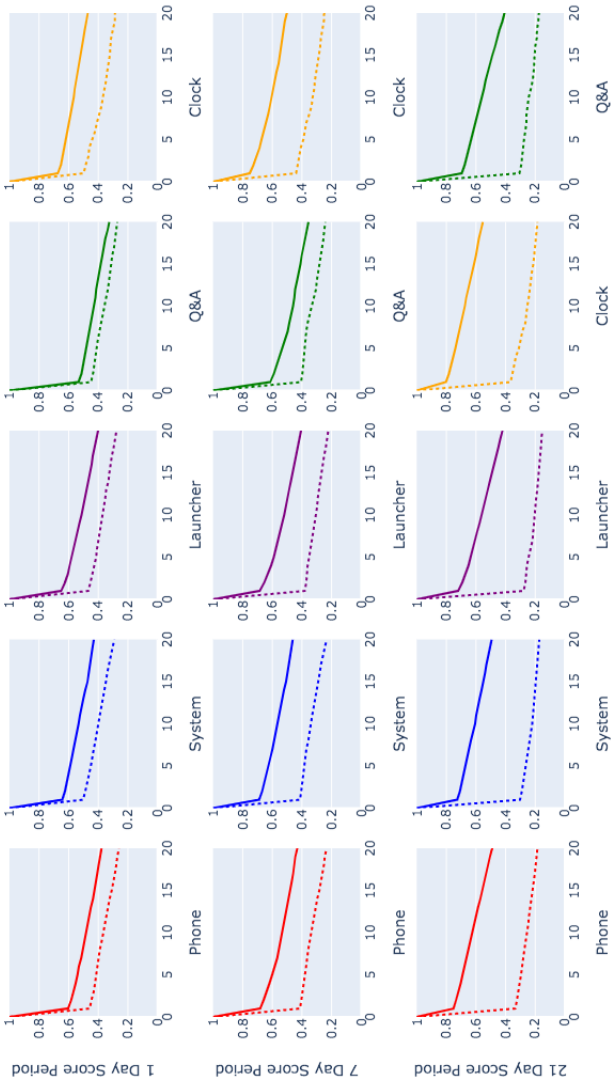

Fig. 4: User Retention per skill is based on the initial scoring duration of 1 day, 7 days, and 21 days. The x-axis is the time in days and the y-axis is the user retention score (the probability of the users returning to Bixby). The solid line in each of the plots indicates the cohort of all users who had a maximum score that was in the top quartile of the overall score distribution for all their utterances. The dotted line in each of the above plots indicates the cohort of all users who had the maximum score in the bottom quartile.

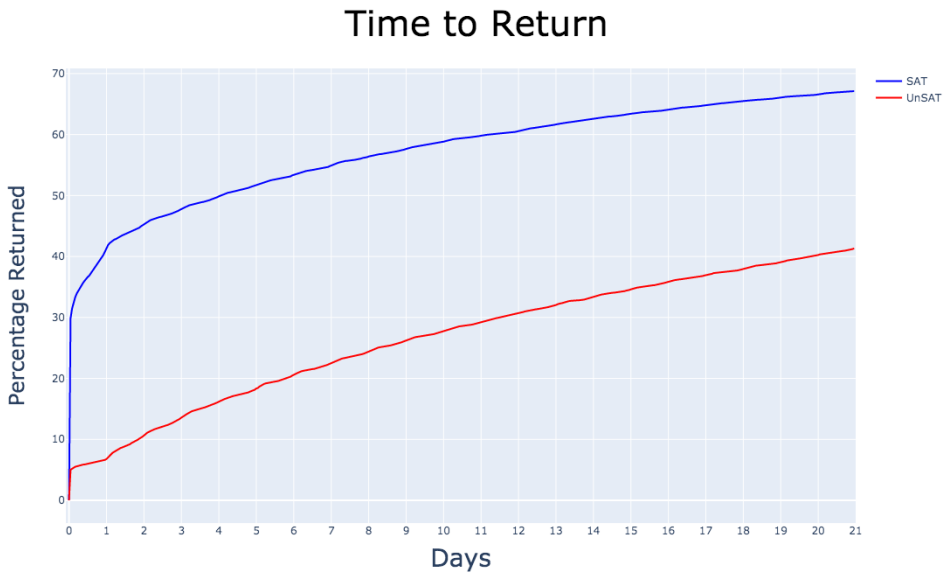

Fig. 5: Time to Return. Percentage of users returning over time after the initial interaction with the system.

7 Conclusions

We presented a machine-learned online task-completion metric that can be used as a surrogate for human graders for measuring user satisfaction rates. Using this predictive metric we showed that bad experiences in the initial period can lead to a sharp drop-off in user retention.

The predictive metric can also be reliably used as a tool to monitor user experience quality for Apps in real-time. This allows us to identify the specific Apps that need improvement and also provides us with specific user utterances with poor Bixby responses, which further enables us to do deep-dive analysis and identify pain points in the end-to-end user experience.

In the future, we plan to expand our work to analyze longer multi-turn dialogue domains that may not have a specific task or goal such as Chat[14]. Metrics for non-task-oriented domains such as Chat can be challenging. Another important dimension is multilingual dialogue since Bixby service is currently also available in other languages like Korean and Chinese [13].

[14] For example, how does one evaluateChatGPT in real-time? This is important since if the conversation is not appropriate, the responsible parties should be notified in real-time. How does it handle current information like the news? What about source attribution? Or “tail” queries?

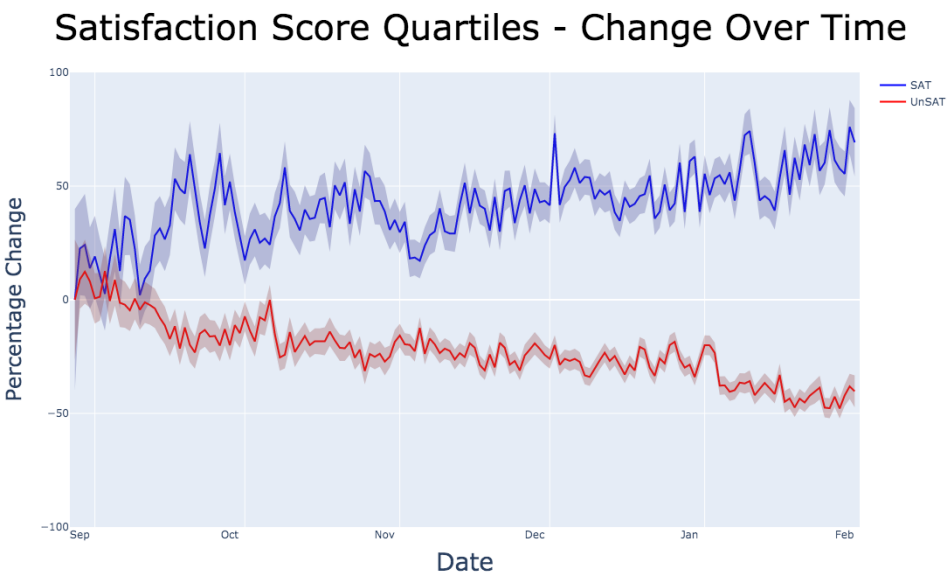

Fig. 6: User satisfaction top and bottom quartiles over time.

Acknowledgement

We sincerely thank the anonymous reviewers for their comments and constructive feedback.

References

1. A. Adhikari, A. Ram, R. Tang, and J. Lin. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*, 2019.
2. L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
3. D. Ciemiewicz, T. Kanungo, A. Laxminarayan, and M. Stone. On the use of long dwell time clicks for measuring user satisfaction – with application to web summarization. Technical Report YL-2010-006, Yahoo Labs, 9 2010.
4. J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
5. M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco. Impact of robot failures and feedback on real-time trust. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 251–258. IEEE, 2013.
6. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

7. J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.
8. S. Gupta, L. Ulanova, S. Bharadwaj, P. Dmitriev, P. Raff, and A. Fabijan. The anatomy of a large-scale online experimentation platform. In *Proceedings of the IEEE International Conference on Software Architecture*, 2018.
9. K. Gwet. Handbook of inter-rater reliability. *Gaithersburg, MD: STATAXIS Publishing Company*, pages 223–246, 2001.
10. S. H. Hashemi, K. Williams, A. E. Kholy, I. Zitouni, and P. A. Crook. Measuring user satisfaction on smart speaker intelligent assistants using intent sensitive query embeddings. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1183–1192, 2018.
11. V. Hu, M. Stone, J. Pedersen, and R. W. White. Effects of search success on search engine reuse. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1841–1846, 2011.
12. C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225, 2014.
13. J.-h. Jhan, Q. Zhu, N. Bengre, and T. Kanungo. $c5l7$: A zero-shot algorithm for intent and slot detection in multilingual task oriented languages. In *Proceedings of the Massively Multilingual Natural Language Understanding Workshop (MMNLU-22)*, pages 62–68, Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022. Association for Computational Linguistics.
14. X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.
15. J. Kiseleva, K. Williams, A. H. Awadallah, A. C. Crook, I. Zitouni, and T. Anastasakos. Predicting user satisfaction with intelligent assistants. In *Proceedings of the 39th ACM SIGIR Conference on Research and Development in Retrieval*, pages 45–54, 2016.
16. J. Kiseleva, K. Williams, J. Jiang, A. H. Awadallah, A. C. Crook, I. Zitouni, and T. Anastasakos. Understanding user satisfaction with intelligent assistants. In *Proceedings of the 2016 ACM Conference on Human Information Interaction and Retrieval*, pages 121–130, 2016.
17. R. Kohavi and R. Longbotham. Online controlled experiments and a/b testing. In *Encylopedia of Machine Learning and Data Mining*. 2016.
18. R. Kohavi, D. Tang, and Y. Xu. *Trustworthy Online Controlled Experiments: A practical guide to A/B testing*. Cambridge University Press, 2020.
19. Z. Li, D. Park, J. Kiseleva, Y.-B. Kim, and S. Lee. Deus: A data-driven approach to estimate user satisfaction in multi-turn dialogues, 2021.
20. Y. Ling, B. Yaho, G. Kohli, T. Pham, and C. Guo. Iq-net: A dnn model for estimating interaction-level dialogue quality with conversational agents. In *Proceedings KDD Converse*, 2020.
21. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
22. R. Meng, Z. Yue, and A. Glass. Predicting user engagement status for online evaluation of intelligent assistants, 2020.
23. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
24. M. Nourani, J. King, and E. Ragan. The role of domain expertise in user trust and the impact of first impressions with intelligent systems. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 8, pages 112–121, 2020.
25. D. Park, H. Yuan, D. Kim, Y. Zhang, S. Matsoukas, Y.-B. Kim, R. Sarikaya, E. Guo, Y. Ling, K. Quinn, P. Huang, B. Yao, and S. Lee. Large-scale hybrid approach for predicting user satisfaction with conversational agents, 2020.
26. M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
27. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
28. V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
29. M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella. Paradise: A framework for evaluating spoken dialogue agents. *arXiv preprint cmp-lg/9704004*, 1997.
30. C.-S. Wu, S. Hoi, R. Socher, and C. Xiong. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Emprical Methods in Natural Langauge Processing*, pages 917–929, 2020.

8 Appendix - Latency Experiments

To compute task completion scores for real-time monitoring, we need a fast hybrid Random Forest. The slowest part of the hybrid Random Forest is the BERT sentence embedding extraction process. We experimented with different types of pre-trained BERT language models to extract embeddings in different dimension sizes for the user and Bixby utterance: DistilBERT-Uncased [28], TinyBERT-6Layer [14], BERTBase-Uncased [6] and TinyBERT-4Layer.

Table 3: Test set latency for different configurations.

Pre-trained model	**Model architecture**	**Mean inference time over 10 runs for one test set sample**
BERTBase-Uncased	12-layer, 768-hidden	22.4 ms
DistilBERT-Uncased	6-layer, 768-hidden	14.61 ms
TinyBERT-6Layer	6-layer, 768-hidden	14.4 ms
TinyBERT-4Layer	4-layer, 312-hidden	3.6 ms

Table 3 shows the test set latency on various configurations. We chose the TinyBERT-4Layer model, which had a 3.6ms latency and 89.25% accuracy on the test set. The corresponding forest parameters were: 200 trees, 40 minimum samples to split an internal node, 40 depth, and 1 minimum sample for the leaf node.