Detecting Absurd Conversations from Intelligent Assistant Logs by Exploiting User Feedback Utterances

Chikara Hashimoto
Yahoo Japan Corporation
Tokyo, Japan
chashimo@yahoo-corp.jp

Manabu Sassano
Yahoo Japan Corporation
Tokyo, Japan
msassano@yahoo-corp.jp

ABSTRACT

Intelligent assistants, such as Siri, are expected to converse comprehensibly with users. To facilitate improvement of their conversational ability, we have developed a method that detects absurd conversations recorded in intelligent assistant logs by identifying user feedback utterances that indicate users' favorable and unfavorable evaluations of intelligent assistant responses; e.g., *"great!"* is favorable, whereas *"what are you talking about?"* is unfavorable. Assuming that absurd/comprehensible conversations tend to be followed by unfavorable/favorable utterances, our method extracts some absurd/comprehensible conversations from the log to train a conversation classifier that sorts all the conversations recorded in the log as either absurd or not. The challenge is that user feedback utterances are often ambiguous; e.g., a user may give an unfavorable utterance (e.g., *"don't be silly!"*) to a comprehensible conversation in which the intelligent assistant was attempting to make a joke. An utterance classifier is thus used to score the feedback utterances in accordance with how unambiguously they indicate absurdity. Experiments showed that our method significantly outperformed methods that lacked a conversation and/or utterance classifier, indicating the effectiveness of the two classifiers. Our method only requires user feedback utterances, which would be independent of domains. Experiments focused on CHITCHAT, WEB SEARCH, and WEATHER domains indicated that our method is likely domain-independent.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; **Discourse, dialogue and pragmatics**; • **Applied computing** → Document analysis;

ACM Reference Format:
Chikara Hashimoto and Manabu Sassano. 2018. Detecting Absurd Conversations from Intelligent Assistant Logs by Exploiting User Feedback Utterances. In *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3178876.3185992

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3185992

Table 1: Example conversations between user and intelligent assistant (IA). First conversation was followed by favorable utterance, indicating that it was likely comprehensible. Second conversation was followed by unfavorable utterance, indicating that it was likely absurd.

User	*From Osaka to Tokyo.*
IA	*Bullet train departing Osaka at 9:00 is available.*
User	*I see, thanks.* ***(Favorable)***
User	*I think I caught a cold.*
IA	*Ha-ha.*
User	*What do you mean?* ***(Unfavorable)***

Table 2: Examples of favorable and unfavorable utterances.

Favorable	Unfavorable
You are smart.	*This is useless.*
Thanks, I got it.	*It doesn't make sense.*
It was a lot of fun.	*I'm gonna uninstall you, bye.*

1 INTRODUCTION

Intelligent assistants (also known as virtual assistants), such as Apple's Siri, have become popular, partly because their natural-language user interfaces simplify access to an enormous amount of information on the web. The key to satisfying users is ensuring that such intelligent assistants converse comprehensibly. To continuously improve intelligent assistant conversational ability, it is important to examine a vast amount of data in a log database in which conversations between users and intelligent assistants have been recorded in order to detect absurd conversations that need to be fixed. Automation of this laborious work would facilitate improvement of intelligent assistant conversational ability.

In this paper, we propose a method for detecting absurd conversations in intelligent assistant logs. The basic idea is simple: absurd conversations tend to be followed by an unfavorable utterance by the user (e.g., *"What are you talking about?"*) and tend not to be followed by a favorable utterance (e.g., *"Great!"*). Table 1 shows two conversations between a user and an intelligent assistant. The first conversation was followed by the favorable utterance *"I see, thanks,"* indicating that it was likely comprehensible. The second one was followed by the unfavorable utterance *"What do you mean?"*, indicating that it was likely absurd.

Favorable and *unfavorable* in this study are used in a broad sense, as shown in Table 2. Favorable utterances include those indicat-

ing, for example, admiration, acceptance, comprehension, appreciation, and amusement. Unfavorable ones typically indicate disappointment, miscommunication, incomprehensibility, contempt, and boredom. We collectively call them *feedback utterances*.

A simple method of detecting absurd conversations would be to match unfavorable feedback utterances against the log entries and extract the immediately preceding conversations (e.g., retrieving the second conversation in Table 1). However, this simple method would face the following two difficulties.

Ambiguity: As is often the case with natural languages, feedback utterances are ambiguous as to which aspect of the intelligent assistant's response the user gave a favorable or unfavorable evaluation. That is, the user may give feedback to not only the overall conversation but also, for example, to the way the intelligent assistant responded (politely or rudely), its speech recognition ability, and the contents (sophisticated or commonplace). For example, a user may give unfavorable feedback (e.g., *"Don't be silly!"*) to a comprehensible response from an intelligent assistant that attempted to make a joke. Furthermore, some users curse at an intelligent assistant for no particular reason on a whim. In contrast, a user may say *"thank you"* after an absurd response that happens to be comforting, such as *"you are wonderful just the way you are"*, even though the response makes no sense in the conversation. In other words, feedback utterances are not always indicative of the absurdity of a conversation.

Infrequency: Users give feedback utterances only occasionally; hence, they are infrequent in the log (about 12% of all user utterances in the log we used in our experiments). Since a significant portion of the absurd conversations may not be followed by a feedback utterance, and simply relying on feedback utterances in the log could result in low recall.

Regarding the ambiguity problem, we observed that some feedback utterances indicate absurdity or comprehensibility more clearly and unambiguously (e.g., *"Your response doesn't make sense."* and *"I see, thanks."*) than others. Therefore, an *utterance classifier* is thus used to score the feedback utterances in accordance with how unambiguously they indicate absurdity. To train the utterance classifier, we prepare labeled data by collecting conversations that were followed by a feedback utterance (e.g., the conversations in Table 1) and labeling the conversations as absurd or comprehensible. From these data, we can learn the strength of association between feedback utterances and absurd conversations, enabling identification of those utterances that tend to unambiguously indicate absurdity.

Regarding the infrequency problem, absurd and comprehensible conversation samples that are followed by a favorable or unfavorable feedback utterances are retrieved. These data are used to train a *conversation classifier*, which sorts conversations, regardless of whether they are followed by a feedback utterance, into absurd or comprehensible. In other words, in our method, absurd and comprehensible conversation samples that are followed by a feedback utterance are not the final detection results but rather are used as labeled data to train the conversation classifier, which can determine the absurdity of *any* conversation even if it is not followed by a feedback utterance. Even if feedback utterances appear infrequently in the log, a large volume of conversation samples can be automatically acquired if the log is large enough. Note that training the conversation classifier requires no manual labor once the feedback utterances are obtained and the log is prepared. Thus, application of this method to a new domain requires only that a log be prepared for the target domain since the feedback utterances are domain-independent in most cases.

We evaluated our basic method and two simplified versions, one without the utterance classifier and the other without the conversation classifier, using five-years' worth of log data for an actual intelligent assistant. The two simplified versions performed much worse than the basic version, indicating that the two classifiers greatly contribute to the performance of our method.

We also compared our method with two baseline methods. One is based on majority voting: whether a conversation is absurd is determined by the number of favorable and unfavorable feedback utterances that follow the conversation in the log. The other is a simple supervised method: the absurdity of a conversation is determined by a supervised classifier trained using manually created conversation data. Obviously, the supervised method requires manual labor for preparing the labeled data for each new domain. Our method substantially outperformed the majority voting method, indicating that simply matching feedback utterances in a log in order to detect absurd conversations works poorly. Our method also outperformed the simple supervised method due to the large volume of automatically acquired training samples.

We evaluated the domain-independence of our method for the three most frequently used domains: CHITCHAT, WEB SEARCH, and WEATHER. We first prepared out-of-domain models that were trained without using training samples for the target domain and were unable to use the feedback utterances that appeared only in the target domain. Then we compared the performance of the out-of-domain and in-domain models using two test sets. There were no statistically significant difference between the models in five of the six settings (three domains $\times$ two test sets), indicating that our method is likely domain-independent.

The contribution of this paper is two-fold: (1) We present a novel method for detecting absurd conversations that exploits user feedback utterances. (2) We empirically demonstrated the effectiveness of our method using a large volume of actual log data.

2 RELATED WORK

The research most related to ours is on how to evaluate an intelligent assistant. Quality metrics can be divided into component metrics and end-to-end metrics [21]. Our method can be seen as an online end-to-end metric, which is further categorized as one based on reference responses [3, 12–14, 18, 23] or one using external knowledge [5, 11, 15, 25]. Reference-based methods typically use the BLEU method [17], which was originally used for the automatic evaluation of machine-translation quality and basically measures word overlap between an intelligent assistant's conversations and reference conversations [3, 12, 13, 18, 23]. Another reference-based method learns to compare reference conversations to an intelligent assistant's conversations [14]. Reference conversations tend to depend on the target domain since the content of reference conversations includes phrases and wording that are domain specific. This means that reference conversations have

to be prepared for each domain. On the other hand, user feedback utterances are generally domain-independent.

With methods using external knowledge, it is assumed that some signals outside the conversation can help identify the absurdity of conversations. Such signals include comments describing the cause of conversational breakdown that are collected from human annotators [5], manually extracted cues [11] and features [15], and positive-reward signals given to correct answers [25]. However, such signals are usually expensive to obtain. User feedback utterances, on the other hand, are likely to occur naturally in conversations with intelligent assistants and are easier to obtain.

Other research directions involving the evaluation of intelligent assistants include predicting user satisfaction with an intelligent assistant. Jiang et al. [7] proposed a method for predicting user satisfaction session-wise. The method proposed by Sano et al. [19] predicts user satisfaction user-wise, i.e., by examining multiple sessions of individual users. Our method evaluates intelligent assistants conversation-wise, which is more useful for detecting absurd conversations, since it would be difficult to pinpoint absurd conversations with session- and user-wise methods. Schmitt et al. [22] proposed a method for predicting interaction quality, an objective measure of user satisfaction, at arbitrary points in a conversation. Since user satisfaction and interaction quality are affected by not only the absurdity of the conversation but also various factors like usability and automatic speech recognition performance [22], we think that methods tailored to user satisfaction prediction are less suitable for absurd conversation detection than our method.

Some studies have used confirmations from users like *"yes"* and *"nope"*, which are a kind of user feedback utterances [7, 11, 15]. Our method uses a wider variety of user feedback utterances like *"I'm gonna uninstall you, bye."* More importantly, we are the first to address the ambiguity and infrequency problems inherent to user feedback utterances, as far as we are aware of.

Reformulation is another kind of feedback utterance that users typically use to correct intelligent assistant speech recognition errors [4, 8, 20, 24], which is outside the scope of this paper.

The use of feedback utterances to automatically acquire the labeled data for the conversation classifier in our method was inspired by distant supervision [6, 16]. In distant supervision, a classifier is trained using automatically collected, weakly labeled samples. To the best of our knowledge, we are the first to apply distant supervision to detecting absurd conversations recorded in intelligent assistant logs.

3 PROPOSED METHOD

Figure 1 gives an overview of our method. In Phase 1, a *scored-feedback-utterance database (DB)* is constructed. It contains favorable and unfavorable feedback utterances that are scored by the utterance classifier in accordance with how unambiguously they indicate absurdity. Feedback utterances to be scored and training data for the utterance classifier are acquired from the log. In Phase 2, absurd conversations recorded in the log are detected using the conversation classifier, which is trained using absurd and comprehensible conversation samples retrieved from the log by exploiting the scored-feedback-utterance DB constructed in the first phase.

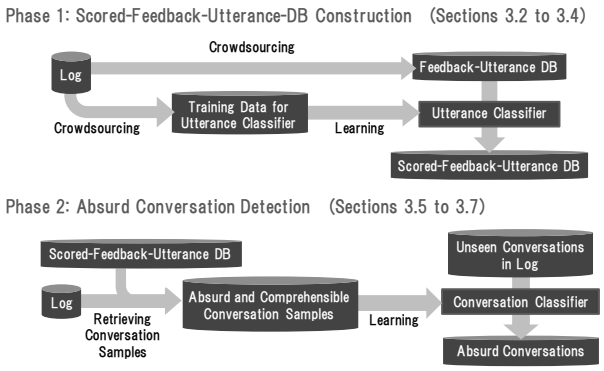

Figure 1: Overview of our method.

Table 3: Examples of conversations between Yahoo! Voice Assist (IA) and its users.

User	*Search the web for Tokyo Skytree.*
IA	*Here is the result of web search for Tokyo Skytree.*
User	*Will it rain tomorrow?*
IA	*The chance of rain tomorrow is 10%.*
User	*I don't feel like going to work today.*
IA	*I know how you feel.*

3.1 Intelligent Assistant Log

The intelligent assistant log used in this study was that of a Japanese commercial intelligent assistant, Yahoo! Voice Assist[1] from April 2012 to May 2017. Yahoo! Voice Assist covers a wide variety of domains, including web search, weather forecasts, chitchat, fortune-telling, sports, and cooking as well as controlling alarms, schedulers, and other applications on smartphones and tablets. Table 3 shows example conversations (web search, weather forecast, and chitchat) between Yahoo! Voice Assist and a user.

We organized the log into sessions, each containing conversations with a single user and not containing two adjacent utterances with an interval exceeding 30 minutes, in accordance with [7].

Although we used a log of Japanese conversations to develop our method, our method is applicable to other languages as well since feedback utterances such as *"your response doesn't make sense."* are common in many languages. The examples were translated into English throughout the paper for illustrative purposes.

3.2 Collecting Feedback Utterances

We first collect feedback utterances, which can be done either manually or automatically. We crowdsourced the labeling, i.e., *favorable*, *unfavorable*, and *neutral*,[2] of 39,435 utterances in the log by using Yahoo! Crowdsourcing.[3] About 20,000 of the utterances were

[1] It has been developed and operated by Yahoo! JAPAN. https://v-assist.yahoo.co.jp
[2] In this study, any user utterance can be given one of these labels. Those that are not considered as feedback like *"what's today's weather?"* are labeled with *neutral*.
[3] It is a service by Yahoo! JAPAN. https://crowdsourcing.yahoo.co.jp

randomly sampled from the log, while the remainder were feedback utterance candidates collected in our preliminary experiments. The utterance length was restricted to less than 30 characters.

Each utterance was labeled by three crowd workers, with the final decision by majority vote. Each utterance was presented to workers without context. The number of workers was 780.[4] The same label was given by the trio of workers to 69.8% of the utterances while only 1.5% were given three different labels, indicating that this labeling task tends to be stable across workers. We discarded those utterances that were given three different labels.

From the crowdsourced labeling results for the 20,000 randomly sampled utterances, we estimated that about 5% and 7% of all the utterances in the log were favorable and unfavorable, respectively.

We then augmented the crowdsourced labeling result with additional feedback utterances (4,066 favorable and 2,434 unfavorable) that we labeled ourselves. Finally, one of the authors checked the augmented labeling result and obtained 20,304 feedback utterances (8,988 favorable and 11,316 unfavorable). We did not include neutral utterances in the feedback-utterance DB.

3.3 Training Utterance Classifier

The utterance classifier takes a feedback utterance as input and outputs a confidence score that the preceding conversation is absurd. We expect that feedback utterances that tend to unambiguously indicate absurdity/comprehensibility are given large/small values and ambiguous ones are given middle values.

3.3.1 Labeled Data. To train the classifier, we prepared labeled data as follows. First, we sampled from the log 38,183 $\langle u, r\rangle$ pairs consisting of a user's utterance (u) and the intelligent assistant's response (r). The u and r were consecutive in a session, and the number of characters was restricted to less than 30 and 150, respectively. We then crowdsourced the annotation of $\langle u, r\rangle$s with labels *absurd* and *comprehensible* to Yahoo! Crowdsourcing. The number of crowd workers was 1,071.[5] Since the absurdity judgment of some conversations requires broader contexts, we allowed the workers to label such $\langle u, r\rangle$s with *uncertain*. Each $\langle u, r\rangle$ was labeled by three crowd workers. 55.7% of the conversations were given the same label by the trio of workers, while only 6.4% were given three different labels, which indicates that this labeling task is reasonably stable across workers. The label was determined to be *absurd* if more than one worker labeled it with *absurd* and *comprehensible* if all three workers labeled it with *comprehensible*. We discarded the other cases. We thereby obtained 22,394 labeled $\langle u, r\rangle$s among which 8,484 (38%) were *absurd* and 13,910 (62%) were *comprehensible*. We split the labeled pairs $\langle u, r, l\rangle$s (l is the label for $\langle u, r\rangle$) into three parts, as shown in Table 4.

From the sessions in the log, we then retrieved feedback utterances for each $\langle u, r, l\rangle$ to prepare tuples $\langle u, r, l, f\rangle$s consisting of a user's utterance (u), the intelligent assistant's response (r), the label (l) for the $\langle u, r\rangle$, and the user's feedback (f) that followed the r. These feedback utterances were those described in Section 3.2. See Table 5 for examples of $\langle u, r, l, f\rangle$s. As a result, we obtained 299,163 $\langle u, r, l, f\rangle$s; the breakdown is shown in Table 6. Note that the numbers are larger than those in Table 4 since some $\langle u, r, l\rangle$s were followed by more than one feedback utterance (f) in the log.

Table 4: Datasets of $\langle u, r, l\rangle$s.

	Set 1	Set 2	Set 3	Total
Absurd	4,787	1,870	1,827	8,484
Comprehensible	7,746	3,056	3,108	13,910
Total	12,533	4,926	4,935	22,394

Table 5: Examples of $\langle u, r, l, f\rangle$s.

u	*From Osaka to Tokyo.*
r	*Bullet train departing Osaka at 9:00 is available.*
l	*comprehensible*
f	*I see, thanks.* ***(Favorable)***
u	*I think I caught a cold.*
r	*Ha-ha.*
l	*absurd*
f	*What do you mean?* ***(Unfavorable)***

Table 6: Datasets of $\langle u, r, l, f\rangle$s. Note that the numbers are larger than those in Table 4, since some $\langle u, r, l\rangle$s were followed by more than one feedback utterance (f) in the log.

	Set 1	Set 2	Set 3	Total
Absurd	77,366	7,907	6,677	91,950
Comprehensible	167,826	19,505	19,882	207,213
Total	245,192	27,412	26,559	299,163

3.3.2 Model . We use fastText[6] with pre-trained word vectors for the utterance classifier due to its speed and accuracy [1, 9]. All hyper-parameters are set to fastText's default values. The word vectors were trained on Japanese Wikipedia articles (as of June 2017) using fastText's skip-gram with all hyper-parameters set to their default values (e.g., the dimension was set to 100). The input to the utterance classifier is an utterance tokenized using the MeCab morphological analyzer.[7] The output is a label, *absurd* or *comprehensible*, and a confidence score associated with the label. The confidence scores s associated with the *comprehensible* label are converted to $-s$ so that the scores all indicate absurdity; the larger the score, the more likely an utterance indicates absurdity. The output is sorted in descending order of the score to draw a precision-recall curve.

Although our method is independent of particular machine learning algorithms and they are not the focus of the current paper, we compare the utterance classifier based on fastText with one based

[4] The number of utterances each worker labeled ranged from 10 to 200. The 780 workers were all considered reliable, since they correctly labeled a small number of utterances for which we knew appropriate labels beforehand. Other workers who failed to label the dummy utterances were counted out. All the workers lived in Japan.

[5] The number of conversations each worker labeled ranged from 5 to 150. All the 1,071 workers correctly labeled a small number of $\langle u, r\rangle$s for which we knew appropriate labels beforehand. Other workers who failed to label the dummy $\langle u, r\rangle$s were counted out. All the workers lived in Japan.

[6] https://github.com/facebookresearch/fastText

[7] http://taku910.github.io/mecab/

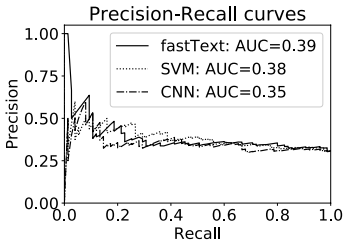

Figure 2: Precision-recall curves for utterance classifiers.

on Support Vector Machine (SVM) [2] and one based on Convolutional Neural Network (CNN) [10] for the purpose of reference.

The SVM-based classifier uses as features a variety of N-grams (surface- and base-form words, word pronunciations and parts-of-speech) with N ranging from 1 to 5.[8] The kernel type and hyper-parameter C, which controls trade-off between training error and margin, were determined so that the area under the precision-recall curve (AUC) was maximized on half of Set 2 of $\langle u, r, l, f \rangle$s. They were set to polynomial $d = 3$ and $C = 1.0$.[9] The output was sorted in descending order of distance from the SVM hyperplane.

The CNN-based classifier was implemented based on Kim's CNN [10].[10] The hyper-parameter settings were the same as those used by Kim [10]; filter windows of 3, 4, and 5 with 100 feature maps each, a dropout rate of 0.5, a mini-batch size of 50, and a dimension of 300 for word vectors (trained on Japanese Wikipedia articles using fastText's skip-gram). The network was trained for 50 epochs, and the network weights obtained for the first epoch were the best with regard to the binary crossentropy loss on the half of Set 2 used to tune the SVM-based classifier. The results were sorted in descending order of the output of the network.

We evaluated the three utterance classifiers using the other half of Set 2 of $\langle u, r, l, f \rangle$s that was not used for tuning the SVM- and CNN-based classifiers. Figure 2 shows the precision-recall curves and the AUC values. The curves indicate that they have similar performances though the one based on fastText was not tuned, unlike the others.

3.4 Scoring Feedback Utterances

Using the utterance classifier based on fastText, we scored the feedback utterances described in Section 3.2 to construct a scored-feedback-utterance DB. Table 7 shows examples of scored feedback utterances with their scores. The scores and the degree to which feedback utterances indicate absurdity were generally correlated.

3.5 Retrieving Absurd and Comprehensible Conversation Samples

Using the scored feedback utterances, we automatically retrieved absurd and comprehensible conversations from the log. Basically, we retrieved $\langle u, r \rangle$s that were followed by feedback utterances with high scores as absurd and those that were followed by feedback utterances with low scores as comprehensible.

Table 7: Examples of scored feedback utterances. Feedback utterances with high/low scores tend to unambiguously indicate absurdity/comprehensibility.

Score	Feedback utterance
0.789062	*It does not make sense.*
0.673828	*I have no idea what you are talking about.*
-0.697266	*Goody, thanks.*
-0.966797	*Well done!*

More specifically, we first retrieved all the $\langle u, r \rangle$s followed by any of the scored feedback utterances from the log. The retrieved $\langle u, r \rangle$s were given the scores of the corresponding feedback utterances. For example, if $\langle u, r \rangle$, (*"I think I caught a cold."*, *"Ha-ha."*), was retrieved and the feedback utterance was *"It does not make sense."* with a score of 0.789062, the $\langle u, r \rangle$ was given a score of 0.789062. Some $\langle u, r \rangle$s were followed by more than one feedback utterance with various scores. They were given the maximum score for the feedback utterances. The retrieved $\langle u, r \rangle$s were then sorted in descending score order. Finally, we took the K $\langle u, r \rangle$s with the highest scores as *absurd conversation samples* and the K $\langle u, r \rangle$s with the lowest scores as *comprehensible conversation samples*. Since the task was to detect absurd conversations, the absurd conversation samples were regarded as positive.

The hyper-parameter K was determined so as to maximize the area under the precision-recall curve for Set 2 of $\langle u, r, l \rangle$s. As a result, $K = 50,000$ was chosen among 10,000, 50,000, 100,000, 150,000, 200,000, 250,000, 300,000, 350,000, 400,000, 450,000, and 500,000.

Table 8 shows examples of absurd and comprehensible conversation samples, feedback utterances that followed the conversations, and scores given to the feedback utterances.

3.6 Training Conversation Classifier

The conversation classifier is trained using the absurd and comprehensible conversation samples. The fastText with pre-trained word vectors is used for the conversation classifier with all hyper-parameters set to their default values. The word vectors are the ones used for the fastText-based utterance classifier. The input to the conversation classifier is a concatenation of u and r of a $\langle u, r \rangle$ segmented by the morphological analyzer. The output of fastText is converted the same way as the utterance classifier as described in Section 3.3.2; the confidence score s for *comprehensible* is switched to $-s$ so that confidence scores indicate absurdity.

3.7 Detecting Absurd Conversations

As the final step, the conversation classifier takes the $\langle u, r \rangle$s in the log as inputs and outputs scores reflecting their absurdity. We applied the conversation classifier to Set 3 of the $\langle u, r, l \rangle$s.

Table 9 shows examples of the classifier's *erroneous* outputs consisting of a $\langle u, r \rangle$, the score given by the conversation classifier, and the label given by the crowd workers. The first example was given a high score, which indicates absurdity, though it is actually comprehensible. This is due to the nature of conversations about

[8] We used SVM-Light (http://svmlight.joachims.org/). For morphological analysis, we used MeCab (http://taku910.github.io/mecab/).
[9] The kernel type was chosen from linear, polynomial $d = 2$, and $d = 3$. C was chosen from 0.01, 0.1, 1.0, 10.0, and 100.0.
[10] We used Keras (https://keras.io/) for the implementation.

Table 8: Examples of absurd and comprehensible conversation samples, feedback utterances that followed the conversations, and scores given to the feedback utterances.

Absurd conversation samples	
u	*What is on your mind?*
r	*My name is Voice Assist.*
f	*It does not make sense.*
Score	*0.789062*
u	*Is it hot this summer?*
r	*I sometimes get hot, too... Sorry.*
f	*I have no idea what you are talking about.*
Score	*0.673828*
Comprehensible conversation samples	
u	*What is* $365 \div 30 \times 12 \times 2 \div 3$*?*
r	*The answer is 97.3333.*
f	*You're smart, aren't you?*
Score	*-0.798828*
u	*From here to Hiraizumi.*
r	*The bus departing at 17:55 is available.*
f	*Great.*
Score	*-0.917969*

Table 9: Examples of the conversation classifier's *erroneous* outputs that consist of $\langle u, r \rangle$, the score given by the classifier, and the label given by the crowd workers.

u	*Tell me a saying.*
r	*The dog that trots about finds a bone.*
Score	*0.964844*
Label	*comprehensible*
u	*Is it cold tomorrow?*
r	*Be careful not to catch a cold.*
Score	*0.544922*
Label	*absurd*
u	*The weather in Rio de Janeiro.*
r	*The weather in Sendai City is snowing and then clear.*
Score	*-0.945313*
Label	*absurd*

sayings; users tend to inquire about the meaning of a saying given by an intelligent assistant by making utterances like *"What does it mean?"*. These utterances tend to resemble unfavorable feedback utterances indicating absurdity. To resolve this problem, we need to distinguish real feedback utterances from utterances that look the same as feedback ones but are actually part of a comprehensible conversation. This will be addressed in future work.

The second example shows a conversation that is not perfectly comprehensible since the intelligent assistant did not answer the question directly. However, the response is at least acceptable as the user can infer or find the answer thanks to the response. This vague and unclear nature is indicated by the mid-range score.

The third example shows a conversation with a low score, which indicates comprehensibility. However, it is in fact absurd since the intelligent assistant gave weather information for the wrong location. Resolving this problem requires commonsense knowledge about locations. Apart from the wrong location, the conversation sounds natural, which would explain the low score.

4 EVALUATION

We evaluated the performance of our method by comparing it with those of baseline methods and variations of our method as described in Section 4.1. We also examined its domain-independence, as described in Section 4.2. First we summarize the results:

(1) Both the utterance and conversation classifiers contribute to the performance of our method.
(2) A method that simply matches feedback utterances against entries in a log to extract absurd conversations has both ambiguity and infrequency problems.
(3) Our method is likely domain-independent.

Preliminary results for our feedback-utterance acquisition method are presented in Section 4.3. This method is well suited for eliminating the manual labor involved in collecting feedback utterances.

4.1 Performance of Proposed Method

4.1.1 Methods Compared. The performances of six methods were compared.

Proposed: Our proposed method.

Proposed-UC: Proposed without the utterance classifier for determining the effectiveness of the utterance classifier. Proposed-UC does not use the scores given to feedback utterances when retrieving absurd and comprehensible conversation samples to train the conversation classifier. Instead, it uses majority vote; it retrieves as absurd conversation samples those that are followed by more unfavorable feedback utterances than favorable ones, whereas conversations followed by more favorable feedback utterances than unfavorable ones are retrieved as comprehensible conversation samples. To be precise, first, each conversation is given a score calculated using $|FB_{unf}| - |FB_{fav}|$, where $|FB_{unf}|$ and $|FB_{fav}|$ are the numbers of unfavorable and favorable feedback utterances that follow the conversation (i.e., large scores indicate absurdity). Then, K conversations with the highest scores and K conversations with the lowest scores are retrieved as absurd and comprehensible conversation samples, respectively. Finally, Proposed-UC trains the conversation classifier using the retrieved conversation samples and scores each conversation the same way as Proposed. The hyperparameter K was set to 50,000 on the basis of Set 2 of $\langle u, r, l \rangle$s.

Proposed-CC: Proposed without the conversation classifier for determining the effectiveness of the conversation classifier. Proposed-CC gives each conversation a score calculated using $|FB_{large}(s)| - |FB_{small}(-s)|$, where $|FB_{large}(s)|$ and $|FB_{small}(-s)|$ are the numbers of feedback utterances that follow the conversation with the utterance classifier scores larger than s and smaller than $-s$, respectively (i.e., large scores indicate absurdity). For conversations without any following feedback utterances, Proposed-CC gives a

Table 10: Performance of all methods. Differences from Proposed are all statistically significant (McNemar's test: $p < 0.01$).

	Accuracy	Precision	Recall	F1 score
Proposed	0.784	0.7158	**0.6907**	0.7031
Proposed+Sup	**0.7998**	**0.751**	0.6869	**0.7176**
Proposed-UC	0.7106	0.6041	0.6338	0.6186
Proposed-CC	0.5277	0.3683	0.3859	0.3769
Supervised	0.7435	0.6735	0.5961	0.6324
MajorityVote	0.5293	0.3685	0.3804	0.3744

very low score (-9,999) since comprehensible conversations are the majority in the log. Since Proposed-CC lacks the conversation classifier, we set threshold t for classification between *absurd* and *comprehensible*. The hyper-parameters s and t were set to 0.5 and -190 on the basis of Set 2 of $\langle u, r, l \rangle$s.[11]

MajorityVote: A baseline that lacks both the utterance and conversation classifiers. MajorityVote simply scores each conversation using $|FB_{unf}| - |FB_{fav}|$, i.e., the same score used in Proposed-UC, and ranks conversations on the basis of this score. Conversations without any following feedback utterances are given a very low score (-9,999). Since MajorityVote lacks the conversation classifier, we set classification threshold t the same way as Proposed-CC ($t = -80$). MajorityVote shows that simply relying on feedback utterances without considering the ambiguity and infrequency problems does not work well.

Supervised: A baseline supervised method. Supervised does not use feedback utterances and the utterance classifier. It simply trains the conversation classifier using Set 1 of $\langle u, r, l \rangle$s. The classifier is based on fastText with the pre-trained word vectors, similar to Proposed. Note that Supervised (and Proposed+Sup below) can be used at the cost of domain-independence since training data for the conversation classifier must be manually created each time this method is applied to a new domain.

Proposed+Sup: A variant of Proposed for determining performance improvement gained by augmenting Proposed with manually created labeled data. The training data for the conversation classifier are the absurd and comprehensible conversation samples (Section 3.5) plus Set 1 of $\langle u, r, l \rangle$s, which is used for Supervised.

4.1.2 Results. Figure 3 shows the precision-recall curves and AUCs for the methods compared. Table 10 summarizes their accuracy, precision, recall, and F1 score. Comparing the performances of Proposed, Proposed-UC, and Proposed-CC, we see that the utterance and conversation classifiers both contributed to the performance of Proposed. The poor performance of Proposed-CC was mainly due to its inability to deal with conversations that are not followed by feedback utterances in the log; it was thus greatly affected by the infrequency problem. Proposed+Sup's performance

[11] The hyper-parameter s was chosen from 0.5, 0.6, 0.7, and 0.8, and t was chosen from $[-200, 200]$ with an increment of 10 (i.e., -200, -190, ..., 190, 200).

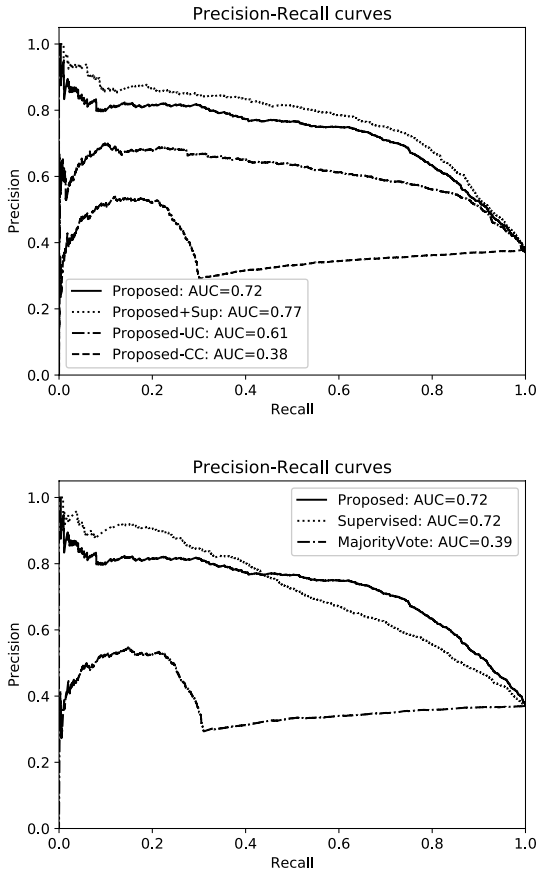

Figure 3: Precision-recall curves and AUCs: Those in top graph are for our method and its variants; those in bottom graph are for our method and baseline methods.

shows that adding Set 1, which was manually created, to the automatically acquired training data improves the performance of Proposed, as expected.

Proposed outperformed Supervised, which we attribute to the automatically acquired training data (Section 3.5), which is more extensive than that of Supervised. MajorityVote performed poorly, mainly due to the ambiguity and infrequency problems.

4.2 Evaluation of Domain-Independence

4.2.1 Experimental Conditions. To evaluate domain independence, we compared out-of-domain and in-domain models of our method. The out-of-domain model was prepared by first training the utterance classifier without training samples, i.e., $\langle u, r, l, f \rangle$s, for the target domain and then constructing a scored-feedback-utterance DB without feedback utterances for the target domain. The domain of $\langle u, r, l, f \rangle$s was determined by our target intelligent assistant, Yahoo! Voice Assist; it classifies each user utterance (u) into a domain (i.e., web search, weather, chitchat, fortune-telling, sports, cooking, and so on), which is assigned

Table 11: Numbers of $\langle u, r, l, f \rangle$s (training samples for the utterance classifier), feedback utterances, and samples for target domain's (Test (Target)) and all domains' (Test (All)) test sets used in domain-independence experiments. Ratios (%) of *absurd* are given in parentheses. Note that the $\langle u, r, l, f \rangle$s and the feedback utterances are for the out-of-domain models; thus, the numbers are for those $\langle u, r, l, f \rangle$s and feedback utterances that do NOT belong to the target domain.

	CHITCHAT	WEB SEARCH	WEATHER
$\langle u, r, l, f \rangle$s	55,573 (9.2%)	223,457 (33.5%)	243,313 (31.7%)
Feedback	10,664	19,025	19,998
Test (Target)	2,274 (66.9%)	1,084 (12.0%)	827 (9.7%)
Test (All)	4,935 (37.0%)	4,935 (37.0%)	4,935 (37.0%)

to the corresponding $\langle u, r, l, f \rangle$. The domain of each feedback utterance (f) in the feedback-utterance DB is similarly determined; each f is assigned the domain of the user utterance in the conversation that f follows in the log.[12] The conversation classifier is then learned in the same way as described in Sections 3.5 and 3.6. In other words, we assume that we know nothing about the target domain in Phase 1 (Figure 1) and that conversations for the target domain are recorded in the log in Phase 2.[13] Note that if we can skip Phase 1, which is the part requiring manual labor in our method, our method can be seen as domain-independent to adapt to a new domain since we can mostly automate domain adaptation. Hyper-parameter K for the out-of-domain model was set to 50,000, i.e., the same value as described in Section 3.5.

The in-domain model was prepared as described in Section 3, except that the number of $\langle u, r, l, f \rangle$s for training the utterance classifier and that of feedback utterances in the scored-feedback utterance DB were reduced to the same numbers as for the out-of-domain model by randomly sampling $\langle u, r, l, f \rangle$s and the feedback utterances, so that the experimental conditions would be the same for the in-domain and out-of-domain models.

We used two kinds of test sets for absurd conversation detection: one was $\langle u, r, l \rangle$s of Set 3, which covers all domains,[14] and the other covered only the target domain. As target domains, we chose CHITCHAT, WEB SEARCH, and WEATHER since they were the most frequent in Set 3. Consequently, we used six settings (three domains × two kinds of test sets).

Table 11 shows the numbers of training samples for the utterance classifier (i.e., $\langle u, r, l, f \rangle$s), feedback utterances, and samples of the two kinds of test sets used for absurd conversation detection. Ratios of *absurd* are given in parentheses. The number of $\langle u, r, l, f \rangle$s for CHITCHAT was smaller than for the other two since CHITCHAT was the most frequent domain, so removing the $\langle u, r, l, f \rangle$s for CHITCHAT would lead to a smaller number of $\langle u, r, l, f \rangle$s. The ratio of *absurd* for CHITCHAT $\langle u, r, l, f \rangle$s was also smaller since many conversations in CHITCHAT were absurd,[15] so removing the $\langle u, r, l, f \rangle$s for CHITCHAT would lead to a smaller ratio of *absurd*. The differences in the number of feedback utterances and in the number of test samples between target domains was due to differences in the frequency of domains. The ratio of *absurd* for a target domain reflects the level of difficulty of having a comprehensible conversation in that domain; it was the most difficult for the CHITCHAT domain, so the test set for the CHITCHAT domain had the largest ratio of *absurd*. In contrast, it was the easiest for the WEATHER domain since the patterns of weather-related utterances are relatively easy to enumerate. In short, absurd conversation detection should be the easiest for the CHITCHAT domain since more than half of the test samples were *absurd*.

We used Set 3 as the all-domains' test set (Test (All), Table 11).

4.2.2 Results. Figure 4 and Table 12 show the results. The difference in performance between the in- and out-of-domain models across the three domains was small; the difference was statistically significant only for WEATHER for the all-domains' test set (McNemar's test: $p < 0.01$) though the difference in precision-recall curves between the two models for the test set was rather small.

These results indicate that our method is likely domain-independent; further investigation using more domains is needed to draw a more definitive conclusion.

4.3 Feedback-Utterance Acquisition

To explore the possibility of automating the collection of feedback utterances rather than using crowdsourcing as described in Section 3.2, we created an utterance-acquisition classifier that sorts utterances into favorable, unfavorable, and neutral.

The training set for the classifier consisted of 9,634 favorable, 10,324 unfavorable, and 9,854 neutral utterances. The test set consisted of 383 favorable, 609 unfavorable, and 8,690 neutral utterances. The test set was sampled from the log mostly randomly, and the training set contained utterances collected in preliminary experiments, which explains why the distributions of favorable, unfavorable, and neutral differed between the two sets.

We used fastText with pre-trained word vectors as the classifier with all the hyper-parameters set to their default values.

We evaluated the classifier for *favorable* and *unfavorable* utterances separately as follows. The *favorable* results were sorted in descending order of the output values for *favorable*. The true labels of the test data were then matched with the classifier's predictions to measure performance. The classifier's *unfavorable* performance was similarly evaluated.

The accuracy and F1 score for *favorable* were 0.9089 and 0.3404, while those for *unfavorable* were 0.9274 and 0.5102. Figure 5 shows the precision-recall curves and AUCs. Given the skewed distribution of the test set, we believe that these results are reasonable and that they can at least serve as a starting point for further study. The worse performance for *favorable* was partly due to utterances containing favorable wording but actually expressing an unfavorable evaluation; e.g., *"Siri is smarter."*

[12] A feedback utterance f can have multiple domains since f can appear multiple times in the log following different conversations. We removed from the scored-feedback-utterance DB those fs that followed only conversations of the target domain.

[13] If we also assume no information about the new domain in Phase 2, we have no clue for learning about the domain and hence no chance of domain adaptation.

[14] The test set was not restricted to the three domains (CHITCHAT, WEB SEARCH, and WEATHER) but was for all the domains assumed in Yahoo! Voice Assist.

[15] CHITCHAT is generally open-ended, so giving a proper response tends to be difficult.

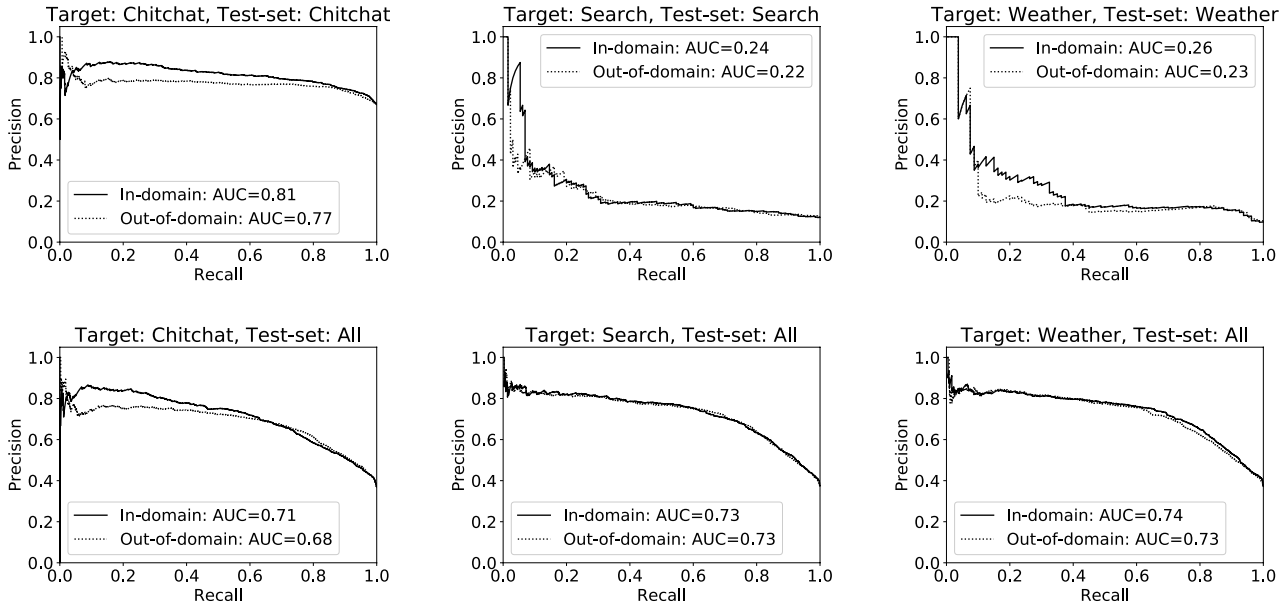

Figure 4: Precision-recall curves: chitchat (left), web search (center), and weather (right). Upper graphs show results for target domain test set; lower graphs show results for all domains test set. In upper graphs, differences in ratios of *absurd* (positive samples) reflect difficulty of having comprehensible conversations for target domain; it was the most difficult for CHITCHAT domain, so the ratio was high (upper left graph). Therefore, absurd conversation detection for CHITCHAT was the easiest since the majority were *absurd*. In contrast, it is easier to converse comprehensibly in WEB SEARCH and WEATHER domains, since they are less open-ended than CHITCHAT, which explains the low ratio of *absurd* (upper right and upper middle graphs).

Table 12: Performance of in- and out-of-domain models. Difference between in- and out-of-domain models were statistically significant only for WEATHER for all-domains' test set (McNemar's test: $p < 0.01$).

(Target domain)		Accuracy	Precision	Recall	F1 score
CHITCHAT	In	0.6988	0.7887	0.7508	0.7693
	Out	0.6812	0.7671	0.7515	0.7592
WEB SEARCH	In	0.7934	0.2169	0.2769	0.2432
	Out	0.7887	0.2171	0.2923	0.2492
WEATHER	In	0.9069	1.0	0.0375	0.0723
	Out	0.9069	1.0	0.0375	0.0723

(All domains)		Accuracy	Precision	Recall	F1 score
CHITCHAT	In	0.7627	0.682	0.6727	0.6773
	Out	0.7611	0.6884	0.6481	0.6676
WEB SEARCH	In	0.7803	0.7027	0.705	0.7038
	Out	0.7868	0.7246	0.6842	0.7038
WEATHER	In	0.7919	0.7307	0.6935	0.7116
	Out	0.7801	0.7032	0.7028	0.703

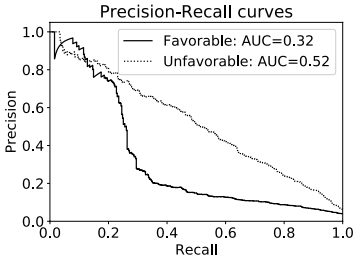

Figure 5: Precision-recall curves for utterance acquisition.

5 CONCLUSION

Our proposed method for detecting absurd conversations with an intelligent assistant exploits user feedback utterances in the log. Experiments showed that our method overcomes the common problems of ambiguity and infrequency by using utterance and conversation classifiers and that it is likely domain-independent.

Since feedback utterances would be common in most languages, we will investigate the language independence of our method. We also plan to release our feedback utterances and the labeled data for utterance and conversation classifiers.

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606* (2016).

[2] Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine Learning* 20, 3 (1995), 273–297.

[3] Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. 2015. deltaBLEU: A Discriminative Metric for Generation Tasks with Intrinsically Diverse Targets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 445–450.

[4] Ahmed Hassan Awadallah, Ranjitha Gurunath Kulkarni, Umut Ozertem, and Rosie Jones. 2015. Characterizing and Predicting Voice Query Reformulation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*. 543–552.

[5] Ryuichiro Higashinaka, Masahiro Mizukami, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, and Yuka Kobayashi. 2015. Fatal or not? Finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2243–2248.

[6] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 3060–3066.

[7] Jiepu Jiang, Ahmed Hassan Awadallah, Rosie Jones, Umut Ozertem, Imed Zitouni, Ranjitha Gurunath Kulkarni, and Omar Zia Khan. 2015. Automatic Online Evaluation of Intelligent Assistants. In *Proceedings of the 24th International Conference on World Wide Web*. 506–516.

[8] Jiepu Jiang, Wei Jeng, and Daqing He. 2013. How Do Users Respond to Voice Input Errors?: Lexical and Phonetic Query Reformulation in Voice Search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 143–152.

[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759* (2016).

[10] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1746–1751.

[11] Emiel Krahmer, Marc Swerts, Mariet Theune, and Mieke Weegels. 2001. Error detection in spoken human-machine interaction. *International Journal of Speech Technology* 4 (2001), 19–30.

[12] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 110–119.

[13] Jiwei Li, Will Monroe, Alan Ritter, Daniel Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep Reinforcement Learning for Dialogue Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1192–1202.

[14] Ryan Lowe, Michael Noseworthy, Iulian V. Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 1116–1126.

[15] Raveesh Meena, Jose Lopes, Gabriel Skantze, and Joakim Gustafson. 2015. Automatic Detection of Miscommunication in Spoken Dialogue Systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 354–363.

[16] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*. 1003–1011.

[17] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. 311–318.

[18] Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-Driven Response Generation in Social Media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 583–593.

[19] Shumpei Sano, Nobuhiro Kaji, and Manabu Sassano. 2016. Prediction of Prospective User Engagement with Intelligent Assistants. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1203–1212.

[20] Shumpei Sano, Nobuhiro Kaji, and Manabu Sassano. 2017. Predicting Causes of Reformulation in Intelligent Assistants. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 299–309.

[21] Ruhi Sarikaya. 2017. The Technology Behind Personal Digital Assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine* 34, 1 (2017), 67–81.

[22] Alexander Schmitt, Benjamin Schatz, and Wolfgang Minker. 2011. Modeling and Predicting Quality in Spoken Human-computer Interaction. In *Proceedings of the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. 173–184.

[23] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. *CoRR* abs/1506.06714 (2015).

[24] Marc Swerts, Diane Litman, and Julia Hirschberg. 2000. Corrections In Spoken Dialogue Systems. In *Proceedings of the Sixth International Conference on Spoken Language Processing*. 615–618.

[25] Jason Weston. 2016. Dialog-based Language Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*. 829–837.