Distribution Agreement

Signature:

________________________________ ______________
Ingyu Jason Choi Date

User Satisfaction Prediction in Open-Domain Conversational Systems

By

Ingyu Jason Choi
Master of Science

Computer Science Department

Eugene Agichtein, Ph.D.
Advisor

Jinho D. Choi, Ph.D.
Committee Member

Surya Kallumadi, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

User Satisfaction Prediction in Open-Domain Conversational Systems

By

Ingyu Jason Choi
B.S., Emory University, 2017

Advisor: Eugene Agichtein, Ph.D.

An abstract of
A thesis submitted to the Faculty of the Graduate School
of Emory University in partial fulfillment
of the requirements for the degree of
Master of Science
in Computer Science Department
2020

Abstract

User Satisfaction Prediction in Open-Domain Conversational Systems
By Ingyu Jason Choi

As voice-based assistants such as Alexa, Siri, and Google Assistant become ubiquitous, users increasingly expect to maintain natural and informative conversations with such systems. For open-domain conversations to be engaging, systems must maintain the user's interest for extended periods, without sounding "boring" or "annoying". Unfortunately, evaluating success and failure remains challenging due to several reasons: (1) open-domain conversations do not have predefined goals; (2) satisfaction is highly subjective to user's preference and system performance; (3) extracting and understanding user behaviors in open-domain conversations are less explored; (4) creating an experiment setting with a functional conversational system requires significant engineering effort.

In this thesis, I proposed a new satisfaction prediction model named ConvSAT that addressed these challenges. First, ConvSAT introduced a new behavioral feature matrix that broke down user behavior and system states into various features, allowing ConvSAT to jointly model heterogeneous signals. Moreover, since many features are generated with direct supervision, measuring feature importance provided a good estimation for identifying positively and negatively correlated behaviors. Second, many previous studies generalized satisfaction prediction problem into offline-setting (prediction after entire conversation) only. However, ConvSAT supports both offline evaluations and online predictions (prediction per each turn), which can be used as live feedback for adaptive dialogue strategies.

I validated the generality of ConvSAT through several applications, implemented as part of the Alexa Prize challenges and Dialogue Breakdown Detection Challenge 3. Lastly, this thesis demonstrates one application of ConvSAT, which is quantifying the effects of modulating prosody (i.e. changing the pitch and cadence of the system response to indicate delight, sadness or other common emotions) on user satisfaction. Together, the results and insights in this thesis provide promising directions for developing a new generation of more responsive and intelligent conversational agents.

User Satisfaction Prediction in Open-Domain Conversational Systems

By

Ingyu Jason Choi
B.S., Emory University, 2017

Advisor: Eugene Agichtein, Ph.D.

A thesis submitted to the Faculty of the Graduate School
of Emory University in partial fulfillment
of the requirements for the degree of
Master of Science
in Computer Science Department
2020

Contents

List of Figures

List of Tables

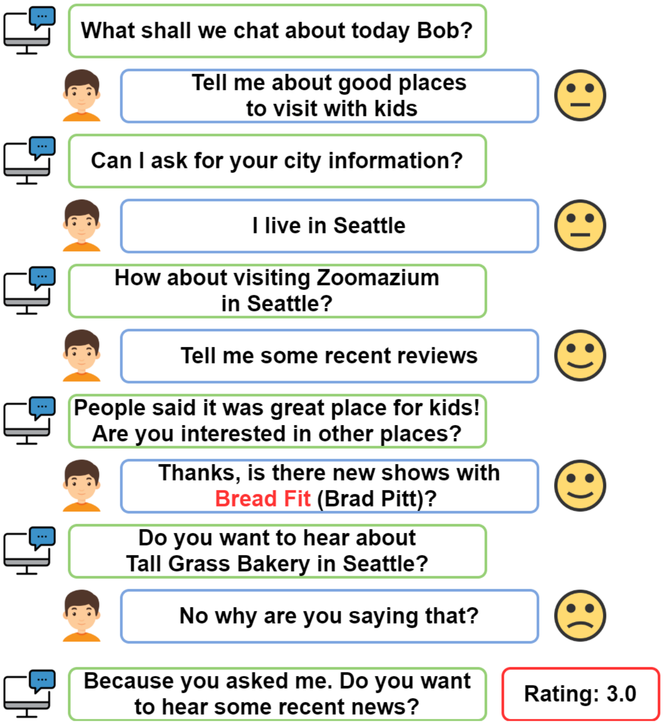
Chapter 1

Introduction

With the proliferation of voice-based assistants such as Alexa and Siri, there has been a resurgence of research into building truly intelligent conversational assistants that can maintain a long, natural conversation with users. Research on conversational AI dates back to the early 60s when researchers demonstrated the potentials of rule-based conversational systems for task-oriented dialogues (i.e. travel assistance or small talk) [8, 56]. These early implementations required extensive human effort since people had to write rules to parse and understand the natural text. More recently, as natural language processing (NLP) and related fields advanced rapidly, conversational systems started to benefit from neural networks trained on a large amount of data [3, 17, 9, 1, 64]. These networks drastically improved the conversation quality as many new models have been proposed for well-known NLP tasks such as text classification and generation [16, 7, 31, 14]. As a result, state-of-the-art (SOTA) conversational systems became extremely popular and versatile tools for a wide range of applications [38, 18].

However, maintaining a coherent open-domain conversation with people is a very challenging task and modern systems still suffer from various failures. As an illustration, consider a sample conversation of a user with our system titled

Irisbot, shown in the Figure 1.1[1]. The conversation started well as *Irisbot* successfully supported multi-turn engagement on travel domain. However, *Irisbot* failed to understand **Brad Pitt** due to automatic speech recognition (ASR) failure, and suggested a local bakery. The user had a hard time understanding the system's non-relevant response, and asked why the system suggested bakery instead of movies. Our system lost context beyond this point, and suggested recent news, as a way of reclaiming the user's interest. At this point, the user was likely dissatisfied, as indeed supported by the 3.0 rating.



What shall we chat about today Bob?

Tell me about good places to visit with kids 😐

Can I ask for your city information?

I live in Seattle 😐

How about visiting Zoomazium in Seattle?

Tell me some recent reviews 🙂

People said it was great place for kids! Are you interested in other places?

Thanks, is there new shows with Bread Fit (Brad Pitt)? 🙂

Do you want to hear about Tall Grass Bakery in Seattle?

No why are you saying that? 🙁

Because you asked me. Do you want to hear some recent news?

Rating: 3.0

Figure 1.1: Sample human-machine conversation from *Irisbot*, with user satisfaction clearly decreasing as the conversation progresses.

[1] Due to the Alexa Prize data confidentiality rules, we cannot reproduce actual user conversations, but the sample represents a typical conversation with our system.

Indeed, to improve conversational systems, understanding the relationship between observable user behaviors and user satisfaction is critical. In Figure 1.1, there are two different types of user satisfaction: (1) offline satisfaction (overall rating of 3.0); (2) online satisfaction (smiley faces immediately after each turn). While offline satisfaction is an overall rating of each conversation, online satisfaction is an intermediate (turn-level) satisfaction, thus equivalent to live feedback. Such distinction is important because many prior studies [24, 33] generalized satisfaction prediction problem into offline setting only. However in real applications, evaluating two different versions of systems based on overall ratings is not practical because the changes can be tiny and ratings can be highly subjective. One of the goals of this thesis is to explore new algorithms that support offline and online satisfaction prediction. Table 1.1 summarizes these two prediction settings.

	Offline prediction	Online prediction
Prediction setting	Every session	Every turn
Allowed context	All context	Observed context so far

Table 1.1: Comparison of two satisfaction prediction settings.

Unfortunately, work on online satisfaction prediction has been limited due to two common reasons: (1) developing an open-domain conversational system for large scale studies requires significant engineering effort; (2) recruiting large-scale users and collecting enough data is challenging. I highlight that the work presented in this thesis is supported by the series of Alexa Prize challenges[2], which provided unlimited computing resources and a large group of Alexa users to make this research possible. Introducing an accurate online satisfaction prediction model would spur dramatic improvements of conversational agents. For example, automatic and timely detection of

[2] https://developer.amazon.com/alexaprize

failures would allow a conversational system to gracefully handle mistakes, and potentially improve both immediate and future system responses. As of now, there have been no methods reported to automatically detect and correct failures as they occur.

To summarize, this thesis will address **three research questions**:

- RQ1: How to identify potential factors that impact user satisfaction in open-domain conversations?
- RQ2: How to train an accurate online & offline satisfaction prediction model for open-domain conversations?
- RQ3: How to apply satisfaction prediction to dialogue evaluation?

To address the first research question, I invested a large amount of time to manually evaluate our system logs during Alexa Prize competition. Previous studies [43, 24] only used textual features (user utterances and system responses) to train satisfaction prediction models, but after internal evaluation, I observed that user utterances in open-domain conversations are much shorter than system responses. Many systems tricked users by indirectly limiting user's choice because preparing responses for open-ended questions was not only challenging but also risky for ratings. To solve this issue, I engineered 51 unique features to represent heterogeneous signals from the natural text, user behaviors, topic preferences and system states. This is one of the main contribution because well-known satisfaction metrics such as clicks, dwell time and touch-based features do not apply to voice-based interaction setting [19, 32, 57, 2]. Lastly, since the engineered features were used to train satisfaction prediction model, analyzing feature importance provided good estimation for identifying strong predictors and understanding user behaviors in voice-based conversations.

For the second and third research questions, I proposed a new conversational satisfaction prediction model (ConvSAT) that supports the prediction of both satisfaction types in a unified architecture. Combined with the handcrafted features, ConvSAT was able to model the complex interaction between user behavior, topic preferences, system states and textual evidence to user satisfaction. ConvSAT is composed of three encoders: (1) contextualized word encoders; (2) contextualized character encoders; (3) behavioral feature encoder. The first two encoders are responsible for learning word-level and char-level token representations while the last encoder learns feature weight of each engineered feature. I highlight that the word-level and char-level representations are learned separately to avoid potential bias toward longer responses. All of these latent representations are fed to a fully connected recurrent network to predict satisfaction of each turn.

For evaluation, I first used an open-source dataset called Dialogue Breakdown Detection Challenge 3 (DBDC3) dataset to verify the empirical effectiveness of ConvSAT. Next, I used a much larger dataset from Alexa Prize 2018 to evaluate in a more realistic setting. At the time of publication (2019), ConvSAT outperformed the strongest state-of-the-art baselines on the benchmark DBDC3 dataset, and achieved about 79% accuracy on classifying satisfaction labels. These results establish that our proposed method not only outperforms the existing state of the art methods, but can successfully generalize to the more challenging real-world scenario of Alexa-based open-domain conversational AI challenge.

Lastly, I showcased one application of using pre-trained ConvSAT to evaluate the change in user satisfaction before and after adding prosody modulation feature to *Irisbot*. Prosody modulation was applied to system responses to express common emotion and avoid monotonous responses. I proposed several metrics to quantify this change from multiple angles. The final results well align with the true ratings from real users, showing a promising direction of using satisfaction prediction models for evaluating unseen dialogues. Together, these results and insights on conversational satisfaction prediction are valuable to the research community and future chatbot designers.

1.1 Summary and Contributions

This thesis presents novel solutions to several research questions defined in Section 1, which are to understand and predict user satisfaction in open-domain human-machine dialogues. First, I report methods on how I extracted 51 unique behavioral features to represent user satisfaction in multiple angles. Then, a new conversational satisfaction prediction model named ConvSAT was proposed to model complex interactions between user satisfaction and heterogeneous signals. Lastly, this thesis demonstrated one application of applying ConvSAT to open-domain dialogue evaluation, specifically to quantify the effects of prosody modulation on user satisfaction. The effectiveness of my proposed methods is evaluated using one publicly available benchmark dataset and one private dataset collected during the Amazon Alexa Prize 2018.

However, there are several limitations to this study. Some of the reported features were tailored specifically to our Alexa Prize implementations. Hence, when applying ConvSAT to datasets not listed in this thesis, it is recommended to retrain ConvSAT with the new group of features. Second, the proposed features were designed for open-domain conversations and may not always apply to goal-oriented settings. Nonetheless, the model architecture and the majority of proposed features can easily generalize across different conversations. The main contributions of this thesis are summarized as follows:

- **A new conversational satisfaction prediction model:** The thesis develops a novel ConvSAT model that leverages conversation context, user behaviors and system-specific states for predicting offline and online user satisfaction for open-domain conversations.

- **A comprehensive list of behavioral features designed to represent open-domain conversations holistically:** The thesis introduces a new set of features tailored to representing user behaviors and system performance in open-domain dialogues. The resulting behavioral feature matrix provides extra guidance for satisfaction prediction, and is applicable to other downstream tasks.

- **An application of ConvSAT to quantify the effects of prosody modulation to user satisfaction:** The thesis showcases an application of utilizing immediate satisfaction predictions to quantify the effects of a newly added feature on user satisfaction. In addition, several new metrics are proposed to evaluate online satisfaction labels from multiple angles.

Together, the results and contributions presented in this thesis are valuable resources for building a new generation of conversational systems that can correct and adapt to failures in real-time.

Chapter 2

Background and Related Work

The belief that humans can interact with machines through conversations has always fascinated people in pop cultures. Not surprisingly, some of the early work on conversational AI predates back to 60s when a researcher named Joseph Weizenbaum from MIT developed *Eliza*, a rule-based chatbot that supports simple chats with people [56]. In this chapter, I summarize how conversational systems evolved and how people attempted to evaluate human-machine dialogues in the past. The works presented in this chapter provide the foundation and give context to the research of this thesis. Some of the materials in this chapter was previously published in these references [12, 3, 54, 11, 4].

2.1 Types of Conversational Systems

Modern conversational systems belong to three main groups: (1) rule-based systems; (2) end-to-end systems; (3) hybrid systems. In this section, I will provide definitions and related literature on each of these groups.

2.1.1 Rule-based Conversational Systems

Rule-based systems solely rely on rules to understand and generate responses. Hence, given the non-deterministic nature of dialogues, these systems suffer from narrow coverage and are not ideal for complex tasks. *Eliza* relied on manually written parsing and response generation rules to interact with people [56]. *Parry* was another popular chatbot developed by Kenneth Colby on 1975, which supported emotion simulation via rule-based scheme [13]. For instance, high anger level of *Parry* could trigger angry responses.

An early project named TRAINS from Rochester University also proposed a multi-turn conversational framework to support planning tasks [5]. These systems had predefined grammar rules and syntactic parsing methods to represent the context. In 1995, a popular chatbot named *Alice* was introduced by Richard Wallace to engage with people on various types of conversations (i.e. small-talk, factoid QA) [52]. *Alice* was heavily inspired by *Eliza*, and utilized Artificial Intelligence Markup Language (AIML) syntax to optimize recursive pattern matching process. *Alice* won the Loebner Prize[1] multiple times on early 2000s, which was the competition to create the most human-like chatbot.

In 2003, a group of researchers from Carneige Mellon University introduced a RavenClaw dialogue framework as a generalized, task-independent framework [8]. This was important because earlier frameworks were tuned heavily to a specific task and were not directly applicable to other tasks without significant engineering efforts. Ravenclaw's architecture suffered less from this drawback because the framework emphasized modular architecture to easily control multiple task-handlers.

[1] `https://en.wikipedia.org/wiki/Loebner_Prize#Contests`

2.1.2 End-to-End Conversational Systems

Unlike rule-based systems, end-to-end systems utilize neural networks to encode observed context into latent space. These latent representations are used as features for two possible setups: (1) response ranking; (2) response generation. A family of models that are trained in response ranking scenarios are retrieval-based models, and the models trained on response generation objective are categorized as generation-based models.

Retrieval-based models are favored when information delivery is required during a conversation, and since the candidate responses are retrieved from the existing corpus, the model is less likely to output grammatically incorrect responses. However, handling out-of-domain utterances is impossible for retrieval-based models when answers are missing from the corpus. On the other hand, generation-based models are more flexible in terms of coverage but are prone to grammatical mistakes and inconsistencies across responses. Generation-based models are also vulnerable from a safe response problem where model only converges to say "safe" responses (i.e. I do not know, I am sorry).

Retrieval-based models

Retrieval-based models are a family of learning-to-rank (LTR) models since the goal is to learn a function to rank a set of candidate responses, prioritized by the response's relevance to the current context. These models typically have a three-staged pipeline: (1) retrieve candidates from corpus; (2) find high-quality matches; (3) re-rank matches [28]. For candidate retrieval, researchers have experimented with various external knowledge bases such as tweets from social media and articles from Wikipedia [10, 61].

For matching, convolutional neural networks are often used for extracting useful features from interaction matrices, which are constructed by computing pair-wise token similarities between tokens in utterances and responses [27, 60]. To incorporate conversation history, sequential matching networks [59] are proposed to model matching signals across current and previous k turns. However, these models had shallower network structure compared to more recent studies that utilize iterated attentive convolution matching [53] to improve matching performance. A slightly different approach [63] leveraged external knowledge through pseudo-relevance feedback and knowledge distillation. Lastly, one recent approach tried to combine retrieval and generation based methods to overcome the drawbacks of each group [62].

Generation-based models

Generation-based models are different from retrieval-based models because the models are trained in a language modeling setting to predict the next likely token, given the current context and previously predicted tokens. Generation halts when the predicted tokens exceed a certain threshold, or outputs an end-of-sentence (EOS) token. Shang et al. [45] proposed a sequence-to-sequence (Seq2Seq) architecture that first encodes given context and uses a decoder to decode compressed information into responses. To alleviate safe response problems, attention mechanism [7, 45] was introduced to learn an alignment function over different tokens. These methods showed promising results because previous approaches compressed all information into one vector and often suffered from information loss [26]. In addition, Maximum Mutual Information (MMI) objective [34] was proposed to improve the diversity of responses.

There have been numerous attempts to ground generation process to specific conditions such as external knowledge and emotional state. For instance, [20] proposed an extended Seq2Seq framework by inserting extra constraint (knowledge representation) vector before decoding. The knowledge vector is obtained by feeding knowledge tokens into a memory network. Similarly, for emotion-grounded responses, an emotion vector is inserted to the decoder through multiple gating mechanisms [55]. Another famous chatbot named *Xiaoice* utilized a similar Seq2Seq structure for generating empathetic responses [64].

More recently, large-scale transformer-based chatbots titled *Meena* [1] and *Blender* [42] were introduced by Google and Facebook, respectively. These systems are similar to the original seq2seq architecture but used transformers as encoders and decoders. Both of these models were trained to minimize perplexity, and showed promising results on training massive end-to-end models for open-domain conversational systems.

2.1.3 Hybrid Conversational Systems

Hybrid conversational systems are different from end-to-end systems because these systems do not rely on one single network to solve everything. Instead, hybrid systems operate on manually written rules to combine outputs from smaller neural networks. For instance, one can define a failure condition to trigger when intent classifier has low confidence, or use a trained ranker to sort responses from multiple rule-based systems [3]. These small networks are all designed to solve specific tasks such as semantic parsing, response ranking and intent classification [4, 29, 63].

The most obvious advantages of hybrid systems are flexibility and reduced maintenance costs. Since hybrid models have no constraint on how to configure multiple models and rules, they can be easily tuned depending on various use cases [3]. However, for rule-based and end-to-end models, re-writing new sets of rules or retraining full model is required to support changes. Hence, many participant teams from Alexa Prize challenges utilized hybrid architecture since the competition lasted live for several months. Motivated by the design from *Ravenclaw* [8], all of the top ranked teams used hybrid architecture with a centralized dialogue manager to support easy configuration of new models and domain handlers [17, 9, 39, 3].

2.2 Conversational System Evaluation

Literature on conversational system evaluation falls into two categories: (1) proposal and analysis of new or existing metrics; (2) user satisfaction prediction.

2.2.1 Previously Proposed Satisfaction Metrics

For goal-oriented conversational systems, *Paradise* framework [50, 51] was introduced back in 90s as a generalized evaluation framework. *Paradise* aims to create a single performance metric by combining multiple smaller metrics. For each dialogue, *Paradise* builds a task-subtask model to track if information delivery was successful. To measure efficiency, dialogue cost was calculated to track the number of turns required to complete a task. *Paradise* framework also supported comparison between sub-dialogues as well as tracking contributions of each metrics to the final performance.

For open-domain conversational systems, additional metrics were proposed since metrics related to task completion became inapplicable. Guo et al. [22] proposed topic-based metrics such as conversational topic depth, topic-specific keyword coverage and topic breadth. Venkatesh et al. [48] proposed a more comprehensive set of metrics such as coherence, engagement and user experience (i.e. ratings) to evaluate systems holistically. Since these metrics are heterogeneous, the authors proposed a stack ranking strategy to simply add all metrics for a final score, or weighted approach with predefined weights.

Several recent studies adopted these metrics to evaluate large-scale social-bots from Alexa Prize teams. Ram et al. [40] discovered that simple combination of these metrics correlate strongly to real user ratings with a 0.66 correlation coefficient. In addition, additional human-annotated metrics such as response quality and response error rate were proposed to improve evaluation criteria [30].

2.2.2 Satisfaction Prediction

User satisfaction can be viewed as an attitude toward an information system, which is measured by various types of beliefs about user interactions as defined in [58, 15, 50, 22, 48]. Satisfaction prediction is different from studies described in Section 2.2.1 because the goal is to train a function that can map observations to user satisfaction. Previously proposed metrics can be used as features when training these models.

For traditional information retrieval (IR) systems such as Web search engines, previous studies showed that incorporating implicit features such as deviations from the average behavior and time on page into the ranking function could improve the search results [2]. For mobile search assistants, combining implicit features with additional touch-related features dramatically increased the performance of a trained satisfaction model [32, 33].

For conversational systems, one recent work proposed a query representation learning technique with intent-sensitive word embeddings, and showed that modifications to improve query representation can improve overall model performance [24]. Another recent work introduced a model that can detect egregious conversations using textual representations, and addressed how this technique can be applied to an automated evaluation scheme [43]. There have been studies to predict causes of query reformulation in intelligent assistants by using system, acoustic, language and additional features [44]. However, all of these work were tested in offline satisfaction prediction setting only.

There have been efforts in restricted domains to predict immediate satisfaction signals, such as using manually curated features from a flight-booking system [46] or detecting online dialogue breakdowns (dissatisfaction) from DBDC3 challenge [25, 36]. Another recent approach proposed a novel self-feeding framework to improve the quality of conversational systems [23] using immediate predictions. I will extend the proposed ideas here by introducing a much more comprehensive set of features to predict both overall & immediate satisfaction in open-domain conversations.

Chapter 3

Conversational Dataset

In this chapter, I present the description of my two data sources: (1) Dialogue Breakdown Detection Challenge 3 (DBDC3); (2) Amazon Alexa Prize 2018. DBDC3 dataset was collected from several chatbots that interacted with real users on small talks. For the Alexa Prize dataset, I used conversational data collected from *Irisbot*, which interacted with thousands of Alexa users. A high-level overview of *Irisbot* is presented as well. Some of the materials in this chapter was previously published in these references [12, 3, 54, 11, 4].

3.1 Dialogue Breakdown Detection Challenge

Dialogue system technology challenges (DSTC), originally known as the dialogue state tracking challenges, were initiated in 2013 to promote research in conversational AI. We focus on the third track of DSTC6'17 challenge titled Dialogue Breakdown Detection Challenge 3 (DBDC3) [25], since it is closely related to online satisfaction prediction. Dialogue breakdown is defined as a situation in conversations where users cannot continue engaging with the system due to various system failures.

3.1.1 DBDC3 Dataset Statistics

Each turn is labeled by 30 human annotators with three labels: 1) not breakdown (NB); 2) potential breakdown (PB); 3) breakdown (B). According to the task specification, turn labels are obtained from majority voting and have to be predicted without looking at future context. We use the official training and test data splits to be consistent with other models published on this data. For our model training, we further set aside 10% of the official training data for model validation. Table 3.1 summarizes the DBDC3 English corpus statistics.

	Training	**Val**	**Test**
Dialogues	373	42	200
Turns	3730	420	2000
NB	1207 (32.3%)	126 (33.3%)	756 (37.8%)
PB	974 (26.1%)	114 (27.1%)	456 (22.8%)
B	1549 (41.5%)	180 (42.8%)	788 (39.4%)

Table 3.1: Dialogue Breakdown Detection Challenge 3 data statistics (English corpus). "NB" stands for not breakdown, "PB" stands for potential breakdown, "B" stands for breakdown.

3.2 Amazon Alexa Prize 2018

This study was also performed as part of a naturalistic assessment of open-domain conversational systems, organized by the Amazon Alexa Prize Conversational AI challenges. Amazon Alexa customers were randomly assigned to each participating system, and could converse on a wide range of topics. At the end of the conversation, the customer could optionally leave a rating

(1.0-5.0) and optional comment feedback. It is worth emphasizing that one of the main goals of the competition was to design an agent capable of maintaining an engaging conversation with a user for 20 minutes, which required significant engineering effort, outlined below, to enable the collection of informative and realistic conversational data. All of the conversations were collected through *Irisbot* interacting with Alexa users.

3.2.1 Irisbot

Our goal was to develop a conversational agent that helps the user be informed about the world around them, while being entertained and engaged. *Irisbot* was developed in a hybrid architecture and incorporated real-time search, informed advice, and the latest information into the conversation by attempting to discuss and share information on many popular domains. To do so, our system had to accurately detect the user's intent from the combinations of explicitly stated and implied evidence from the context. The detailed description of the agent architecture, dialogue manager, retrieval modules and response ranking is illustrated in Figure 3.1.

Centralized Dialogue Manager

IrisBot is designed through a loose coupling of domain-specific retrieval modules that interact through centralized dialogue manager. Dialogue manager is responsible for first receiving transcribed audio input from automatic speech recognition (ASR) module. These transcribed utterances are processed through the NLP/NLU modules to identify the key entities, topics, intents and other helpful NLP features. Dialogue manager maintains a context object to store all dialogue states as well as NLP/NLU outputs. Context object is used for retrieval modules to generate candidate responses, which are later ranked by our global ranker. The best response is converted back

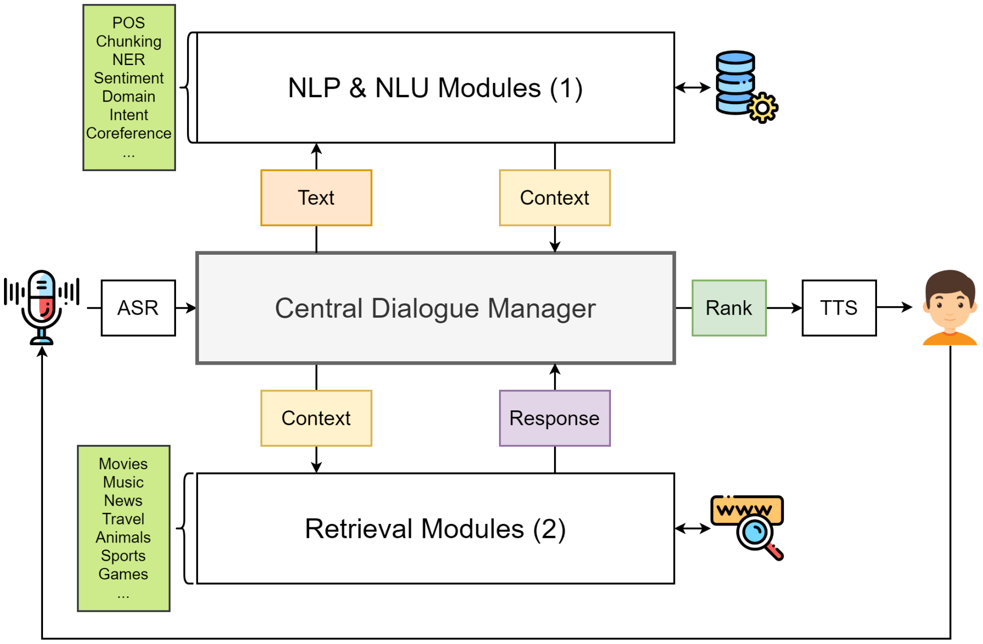


Figure 3.1: Architecture of Irisbot, an open-domain socialbot developed during Amazon Alexa Prize 2018. Each utterance is first processed through our NLP/NLU pipeline, followed by candidate response generation from multiple retrieval modules. Candidate responses are ranked based on their estimated relevance to the current context and system states. (1) and (2) indicates the order of data flow.

to audio signal through a text-to-speech (TTS) module, and waits for the user's input.

NLP/NLU Modules

Irisbot utilized a variety of NLP models to extract useful features. Each utterance was sent to all of our modules in parallel to reduce latency. These modules include POS tagging, chunking, sentiment analysis, domain/intent classification, named entity recognition and coreference resolution. For POS tagging and chunking, we used a pretrained classifier from open-sourced NLTK library [35], and sentiment analysis was performed using Vader [21]. Our domain/intent classifier [4] leveraged mixture-of-expert models and topic transition matrix to predict most likely labels given user utterance and previous context. For named entity recognition, we used multiple knowledge bases (i.e. DBPedia, Wikipedia) to lookup candidates via soft n-gram matching [6]. Lastly, coreference resolution was done based on heuristics.

Retrieval Modules and Response Ranking

We provide brief descriptions of some of the most popular domain-specific retrieval modules. The full list of our retrieval modules can be found in [3].

- *Opening*: Introduction begins with a required greeting to identify the agent as a specialized Alexa skill, and attempts to "break the ice" with the user by exchanging names, and proposing initial topics for discussion.

- *Movies*: Movies retrieval module can hold in-depth conversations on most movie-related topics including trending movies, TV shows, actor/director information and personalized movie recommendations.

- *Music*: Music retrieval module handles popular music-related questions such as trending chart by genre, upcoming concert information and music recommendations.
- *News*: News retrieval module is responsible for updating the customer with trending news or news on specific entities. It covers a wide range of popular news domains such as politics, science, celebrity, sports and so on.
- *Games*: Games retrieval module can chat and recommend the most popular upcoming games for various gaming platforms such as PlayStation, Xbox and PC.
- *Travel*: Travel retrieval module supports real-time place searches such as retrieving recent reviews, ratings and addresses.

We chose to do “lazy” response evaluation in that the final response ranking is performed after each module returns a candidate response, at which point the responses are ranked and selected based on the estimated relevance. Thus, each query is processed by every retrieval module in parallel. Each domain retrieval module implements a common set of interfaces, and is expected to return a score of the response, the response type and topic, and follow-up suggestion (which could be the same retrieval module or a switch to another topic). As a result, adding new retrieval modules turned out to be quite easy with the main challenge being to expand the topic classifier to identify when the new retrieval module is relevant.

3.2.2 Alexa Prize Dataset Statistics

Throughout several months of competition, *Irisbot* collected about 20,000 rated conversations. These ratings are obtained once conversations were over, thus are equivalent to offline satisfaction labels. I highlight that because our system was constantly updated, the quality of conversations differ significantly throughout this duration. Hence, I will only focus on conversations from one stable version of our system, with the data collected over the last 2-weeks period in August 2018. The data used for this study contained 5,044 rated conversations, with 4,811 conversations (95.3%) from unique users. We randomly selected 93 conversations as our test set, and selected an additional 10% of the remainder as our validation set for training. Table 6.1 reports the statistics for Training, Validation, and Test data splits.

	Training	**Val**	**Test**
Dialogues	4455	496	93
Turns	80996	8864	1959
$\mathbf{Turns_{avg}}$	18.18	17.87	21.06
$\mathbf{Rating_1}$	593 (13.3%)	62 (12.5%)	10 (10.7%)
$\mathbf{Rating_2}$	671 (15.0%)	74 (14.9%)	11 (11.8%)
$\mathbf{Rating_3}$	811 (18.2%)	95 (19.1%)	17 (18.2%)
$\mathbf{Rating_4}$	860 (19.3%)	96 (19.3%)	19 (20.4%)
$\mathbf{Rating_5}$	1520 (34.1%)	169 (34.0%)	36 (38.7%)

Table 3.2: Alexa Prize 2018 data statistics.

For the entire data, the standard deviation on turns is **15.81**, meaning our data covers a wide range of different conversations from extremely short, to very long ones, with some conversations lasting over 100 turns. Interestingly, there was no strong correlation between a user rating and conversation length: the Pearson correlation coefficient is **0.095**, indicating no correlation. Lastly, our system supports conversations on 15 different domains, ranging from popular domains such as *Movies* and *Music* to generic domains such as *Weather* and *Wikipedia*. Our domain classifier, described in reference [4] achieved **0.717** Micro-Averaged F1 on our 3,000 annotated test utterances.

3.2.3 User rating vs. user satisfaction

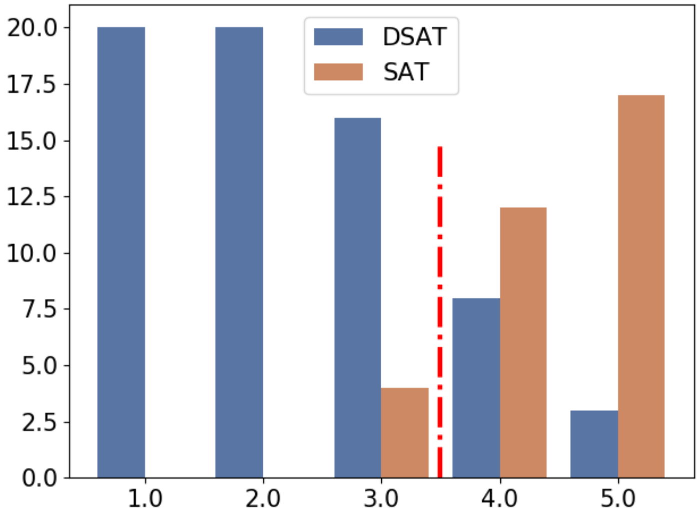


Figure 3.2: Count (y-axis) of dissatisfied and satisfied feedback among different rating groups (x-axis). The red line indicates the best cut (rating=3.5) between SAT/DSAT labels.

User rating and user satisfaction are clearly related, but they are different metrics. In an open-domain setting, user ratings can be highly subjective and cannot generalize to five-point scale rating system. For instance, rating 3.0 can mean mediocre or terrible performance depending on raters. Hence, I utilized user feedback, a free-form optional feedback from a subset of users, to find a statistical relationship between rating and satisfaction. I randomly selected 20 feedback each from five rating groups and asked one human annotator to label each feedback as satisfied or dissatisfied. The goal was to find a rating threshold that best splits satisfaction (SAT) and dissatisfaction (DSAT). There is a long tradition in evaluation literature for this approach, e.g., [33, 23, 32, 43, 24] in order to reduce high subjectivity and noise in user ratings. The challenge is where to choose the boundary to convert the user ratings to SAT/DSAT decisions. The annotation results are reported in Figure 3.2.

The annotation results indicated that for 1.0 and 2.0 rating groups, 100% of users left negative feedback based on their interactions. For the 3.0 rating group, I saw a small increase in positive feedback, but still, 80% of users were dissatisfied. For 4.0 and 5.0 rating groups, only 40% and 15% of users were dissatisfied. Hence, I concluded that setting a boundary between 3.0 and 4.0 ratings will best separate dissatisfaction from satisfaction, and I defined the two user satisfaction labels as DSAT (ratings $<=$ 3.5) and SAT (ratings $>$ 3.5). Defining SAT to correspond to ratings of over 3.5 out of 5 has an additional benefit. One important goal of online satisfaction prediction is to provide consistent and reliable reinforcement signals for tasks such as online dialogue policy learning or model tuning. For such tasks, knowing highly satisfactory (and strongly dis-satisfactory) outcomes is valuable, while intermediate “partially” satisfied signals are not helpful.

3.2.4 Annotating online satisfaction labels

I reduced rating prediction problem into a binary classification problem based on the user feedback analysis. However, I emphasize that user ratings were requested after the conversation ended, and do not provide online satisfaction labels. To obtain these ground truth labels, I asked two human annotators to label 1,959 turns using the annotation guidelines below. Only the conversation transcripts data (utterances and responses) were provided during the annotation process.

- *Label each turn into SAT or DSAT by considering all the previous information up to the current turn.*

- *Factors to consider are conversational depth within the current topic, conversational coherency, domain detection rate, response quality, topic diversity, ASR and other miscellaneous errors.*

For offline predictions, I used the satisfaction label derived from real ratings. Hence, the number of offline samples (93) is identical to the number of dialogues (93) as shown in Table 3.2. The final SAT class distribution of offline and online test samples is 40.9% and 56.8% respectively. The kappa score [49] between the two annotators on these 1866 samples is **0.753**, showing a substantial agreement. In the case of a disagreement, the final label was randomly chosen.

Chapter 4

ConvSAT: Conversational Satisfaction Prediction

4.1 ConvSAT: Method Description

In this chapter, I present my proposed conversational satisfaction prediction model (ConvSAT). As illustrated in Figure 4.1, ConvSAT considers three complementary input dimensions: 1) contextualized word encoders (colored green); 2) contextualized character encoders (colored blue); 3) behavioral feature matrices (colored yellow). The following descriptions are referenced from the original paper [12].

4.1.1 Model Architecture

Contextualized Word Encoders To add context history, I defined a hyper-parameter called context window size (W) to control how many previous turns to condition. To ensure an online setting, I did not incorporate any future information. Hence, given previous turns (T_1 ... $T_{i\text{-}1}$) and current turn (T_i), current utterance (U_i) and current response (R_i) were expanded with previous W turns. I fixed W=3 for the illustration purpose throughout this section.

$$U_i = [U_{i-3}; U_{i-2}; U_{i-1}; U_i] \tag{4.1}$$

$$R_i = [R_{i-3}; R_{i-2}; R_{i-1}; R_i] \tag{4.2}$$

The boundaries between the expanded utterances and responses are marked with special tokens. These two expanded sequences are tokenized to obtain two word sequences ($U_i{}^w$, $R_i{}^w$), which will be the inputs to contextualized word encoders:

$$U_i{}^w = [U_{w1}; U_{w2}; U_{w3}; U_{w4} \text{ ... } U_{wn}] \tag{4.3}$$

$$R_i{}^w = [R_{w1}; R_{w2}; R_{w3}; R_{w4} \text{ ... } R_{wn}] \tag{4.4}$$

To represent the utterances contextually, I chose bidirectional Long Short Term Memory (bi-LSTM) networks, as they have shown promising performance for representing text. I used two separate encoders for both utterances ($Encoder_U$) and responses ($Encoder_R$). This is because in human-machine conversations, the ratio of words in an utterance to response is low, mainly due to limitations in open-domain conversational systems. By using two separate encoders, the goal was to reduce the possible bias towards long responses. The last hidden outputs from each forward LSTM ($\overrightarrow{h_n}$) and backward LSTM ($\overleftarrow{h_n}$) were concatenated to represent the entire word semantics in U_i and R_i. These two outputs were concatenated to obtain the final context representation ($Encoder_{word}$) at T_i:

$$Encoder_{\text{word}} = [Encoder_U; Encoder_R] \tag{4.5}$$

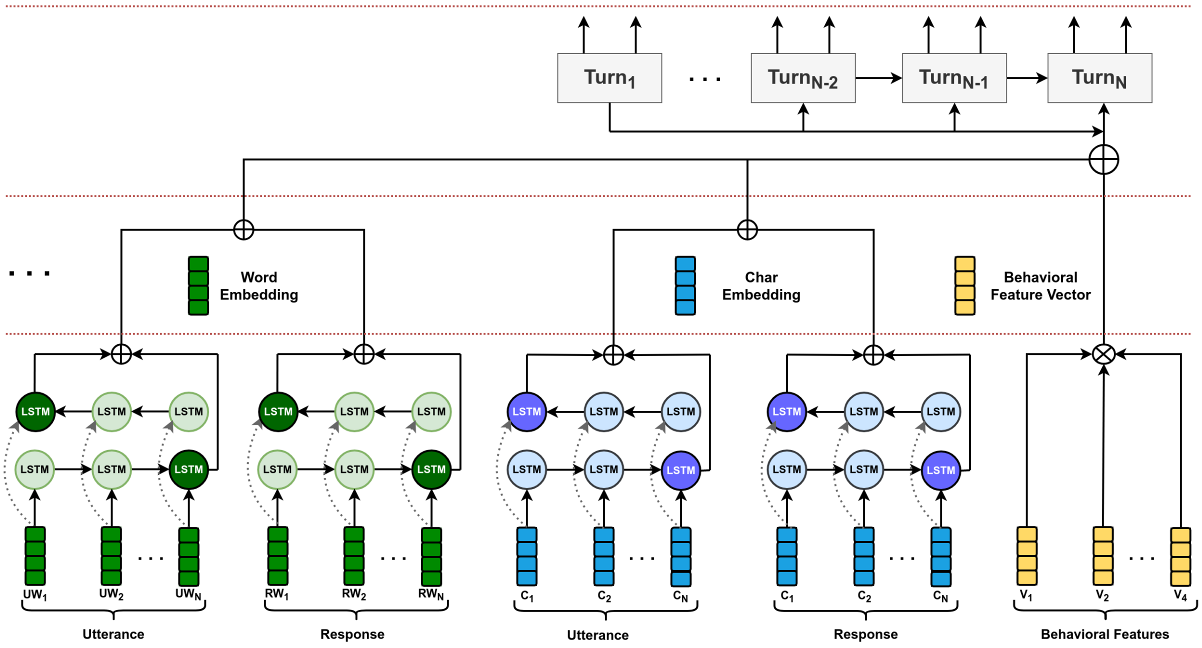


Figure 4.1: Model architecture of ConvSAT (best viewed in color).

Contextualized Character Encoders Voice-based conversational systems are vulnerable to automatic speech recognition (ASR) errors. Errors were more frequent for entity names, such as people or brand names, and transcription errors in these often resulted in a failed conversation. I noticed that mis-spelled or mis-segmented words often shared similar sub-word structures, because various accents and pronunciations originated from a single root word. As an illustration, consider a short example of how ASR recognized several automobile brands for people with foreign accents:

Actual word	**ASR failures**
Mercedes	Sadis, Cedes, Sadi's
McLaren	Mac Laren, Mac Lauren, Mclaurin
Aston Martin	Astone Martine, Ask Tony Martin

Without subword (character-level) information, these errors are likely to create noise in learning robust word representations. Moreover, the frequency of errors such as Sadi's appearing in our data is low, which causes the embedding matrix to be more sparse. For the Ask Tony Martin case, it is likely that the model will understand this phrase differently from the original intent. Hence, by jointly training word-level and sub-word (character-level) models, I hypothesize that the overall semantics can be modeled better.

From the expanded word sequences $U_i{}^w$, $R_i{}^w$ in (3) and (4), I derive the character sequences $U_i{}^c$ and $R_i{}^c$:

$$U_{\mathrm{i}}{}^{\mathrm{c}} = [[c_{1,1} \; ... \; c_{1,k}]; [c_{2,1} \; ... \; c_{2,k}] \; ... \; [c_{n,1} \; ... \; c_{n,k}]] \tag{4.6}$$

$$R_{\mathrm{i}}{}^{\mathrm{c}} = [[c_{1,1} \; ... \; c_{1,k}]; [c_{2,1} \; ... \; c_{2,k}] \; ... \; [c_{n,1} \; ... \; c_{n,k}]] \tag{4.7}$$

The following $U_i{}^c$ and $R_i{}^c$ are 2-dimensional matrices with first dimensions representing each tokenized word and second dimensions representing characters of each word. I flatten these matrices to two 1-dimensional character sequences. I also used bi-LSTM networks ($\mathrm{Encoder}_{\mathrm{Uc}}$, $\mathrm{Encoder}_{\mathrm{Rc}}$) to obtain final character representation ($Encoder_{char}$), which is identical to the process in (5).

Behavioral Features with Online Scaling Behavioral features are manually engineered to encode different aspects of user behavior. At a particular turn T_i, user behavior is represented as one feature vector (v_i), which can be a concatenation of various types of features. To incorporate conversational context, I append last W feature vectors to obtain matrix V_i:

$$V_{\mathrm{i}} = [v_{i-3}; v_{i-2}; v_{i-1}; v_i] \tag{4.8}$$

Each v_n encodes local information from beginning turn T_0 to turn T_n. For instance, if I count total words in current T_i, total words are counted from T_0

to T_i. Similarly, when computing the average number of words, total words from T_0 to T_i is divided by the current turn i. Our proposed scaling function $S(v,\ i)$ scales feature vectors (v) with respect to the current turn index (i). For online predictions, such scaling mechanism is crucial, because the goal is to detect a relative change in user behavior as the conversation progresses. An illustration of online scaling function is presented in Figure 4.2.

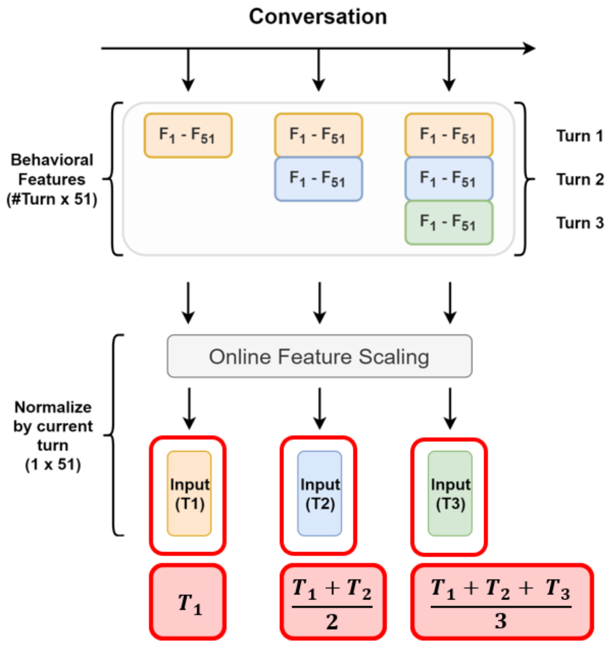


Figure 4.2: Visualization of proposed online scaling function. Each feature vector ($\mathrm{T_n}$) is normalized based on the length of the observed turns.

For instance, if a user engaged deeply in one topic but started to diverge in the later turns, a feature capturing topic transition rate (how likely conversational states change) will gradually increase from lower to higher values. I apply this online scaling function to each vector in V_i to obtain scaled $\hat{V}_{\mathrm{i}}$:

$$\hat{V}_{\mathrm{i}} = [S(v_{i-3}, i-3),\ S(v_{i-2}, i-2)\ ...\ S(v_i, i)] \tag{4.9}$$

The resulting $\hat{V}_{\mathrm{i}}$ is a 2-dimensional dense matrix, with row representing each turn and column representing each scaled feature in respect to that turn i. Then, I feed $\hat{V}_{\mathrm{i}}$ to an attention layer to obtain a weighted sum of each vector. Given each v_i, similarity score $\mathrm{s_i}$ is computed based on a shared trainable matrix M, feature context vector c and a bias term b_i. M, c and b_i are initialized randomly and jointly learned during training. Softmax activation is applied to similarity scores to obtain attention weights α. Lastly, using learned α, each v is multiplied to its attention weight α_{i} and summed to obtain the attended output $\hat{V}_{\mathrm{i}}{}^{\mathrm{att}}$:

$$s_i = tanh(M^{\mathrm{T}} v_i + b_i) \tag{4.10}$$

$$\alpha_i = \frac{exp(s_{\mathrm{i}}{}^{\mathrm{T}} c)}{\sum_{i=1}^{W} exp(s_{\mathrm{i}}{}^{\mathrm{T}} c)} \tag{4.11}$$

$$\hat{V}_{\mathrm{i}}{}^{\mathrm{att}} = \sum_{i=1}^{W} \alpha_{\mathrm{i}} v_{\mathrm{i}} \tag{4.12}$$

This is equivalent of learning how much previous information to attend when modeling relative changes in user behaviors by learning the weight of each turn.

Fully Connected Layer The outputs from contextualized word encoders, char encoders and attended feature matrix are concatenated to obtain each turn representation:

$$Turn_{\mathrm{i}} = [Encoder_{\mathrm{word}};\ Encoder_{\mathrm{char}};\ \hat{V}_{\mathrm{att}}] \tag{4.13}$$

To benefit from all previous turn outputs, I have one final unidirectional LSTM that models each turn sequentially. Depending on tasks (online or offline prediction), many-to-many or many-to-one output(s) can be obtained.

Each output is fed to a linear layer with dropout to enforce regularization, followed by sigmoid or softmax activation to obtain binary or multi-class distribution.

4.1.2 Behavioral Features

Behavioral features extracted for ConvSAT are categorized into three types: 1) general behavioral features; 2) system features; 3) topic preference features. These features are concatenated to produce one feature vector per each turn.

General Behavioral Features General behavioral features are features that encode user behaviors in various dimensions, including lexical, semantics and conversational. First, I define engagements as subsets of conversation that have 4+ conversational depth on the same topic. Count of engagements (F_1) and max length of engagements (F_2) are derived respectively. Sentiment analysis using Valence Aware Dictionary for sEntiment Reasoning (VADER) [21] on utterances is applied to obtain positive (F_3, F_5) and negative (F_4, F_6) sentiment scores. To capture how much topic transition occurs, state change ratio (F_7) is derived by dividing total transitions to the current turn index. Similarly, agreement and disagreement ratios are derived (F_8, F_9) based on intent classification results. To measure the repetition between (U_i, R_i), (R_{i-1}, R_i) and (U_{i-1}, U_i), counts of token overlaps are computed (F_{10}, F_{11}, F_{12}). Lastly, the average and total word count of user utterances and system responses are extracted (F_{13} ... F_{18}).

System Features System features are directly related to systematic aspects of our conversational agent. There are two binary session-level features that capture if a user agreed to provide his name or if he is a returning user (F_{19}, F_{20}). For latency, I define two types, which are system latency

Local Features	**Short Description**
F_1 - *NumEngagements*	#Engagements
F_2 - *MaxEngagements*	Max engagement in # of turns
F_3 - *UtterancePos*	Positive sentiment in U_i
F_4 - *UtteranceNeg*	Negative sentiment in U_i
F_5 - *AvgPos*	Sum of pos sentiment counts / i
F_6 - *AvgNeg*	Sum of neg sentiment counts / i
F_7 - *StateChangeRatio*	#Topic Transitions / i
F_8 - *YesRatio*	#Yes Responses/Agreements / i
F_9 - *NoRatio*	#No Responses/Disagreements / i
F_{10} - $TokenOverlap_U$	Token overlap in U_i, $U_{i\text{-}1}$
F_{11} - $TokenOverlap_R$	Token overlap in R_i, $R_{i\text{-}1}$
F_{12} - $TokenOverlap_{UR}$	Token overlap in U_i, R_i
F_{13} - $TotalWord_U$	Total #Words in U_i
F_{14} - $TotalWord_R$	Total #Words in R_i
F_{15} - $AvgWord_U$	Average #Words in U_1 ... U_i
F_{16} - $AvgWord_R$	Average #Words in R_1 ... R_i
F_{17} - $Word_U$	#Words only in U_i
F_{18} - $Word_R$	#Words only in R_i

(F_{21}, F_{22}, F_{23}) and user latency (F_{24}, F_{25}, F_{26}), both measured in seconds. System latency measures how long a user had to wait to hear the system response; user latency measures how long a user had to think before issuing an utterance. Lastly, every token in our utterances was annotated with ASR confidence value ranging from 0.0 to 1.0. Using these values, minimum, maximum and average token confidence on each U_i are added (F_{27}, F_{28}, F_{29}).

Session-level Features	**Short Description**
F_{19} - *NameProvided*	Name provided or not
F_{20} - *ReturningUser*	Returning user or not
Local Features	**Short Description**
F_{21} - *Latency*	System latency on U_i
F_{22} - $Latency_{avg}$	Average system latency
F_{23} - $Latency_{max}$	Max system latency
F_{24} - *UserLatency*	User latency on R_i
F_{25} - $UserLatency_{avg}$	Average user latency
F_{26} - $UserLatency_{max}$	Max user latency
F_{27} - ASR_{min}	Min token confidence on U_i
F_{28} - ASR_{max}	Max token confidence on U_i
F_{29} - ASR_{avg}	Average token confidence on U_i

Topic Preference Features Topic distribution features encode specific behaviors related to topic diversity, visited topics and topic distribution so far. For topic diversity, I counted the length of the visited topic set to represent topic breadth (F_{30}). Count of accepted topics and rejected topics (F_{31}, F_{32}) are extracted to explore topic acceptance and rejection trade-offs. Lastly, a 15-dim topic count vector and a 3-dim special state count vector from T_0 to T_i are concatenated to represent the online topic distribution (F_{33}, ... F_{51}). The special states include Stop, Profanity and Clarification. Stop state tracks whether a user expressed stop signals, profanity state tracks if an utterance or response contained profane words, clarification state tracks if system asked a user to repeat due to low ASR confidence.

Local Features	**Short Description**
F_{30} - *TopicBreadth*	Number of unique topics visited
F_{31} - *TotalAcceptedTopics*	#Accepted topics
F_{32} - *TotalRejectedTopics*	#Rejected topics
$F_{33...51}$ - *TopicDistribution*	Vector of 18 topic counts

4.1.3 Additional Implementation Details

For contextualized word encoders, embedding weights are initialized with pretrained Google Word2Vec [37] of size 300 and tuned for conversational context. For contextualized char encoders, embedding weights of size 32 are randomly initialized and learned during training. I used 3 for W, since I observed adding less or more context reduced performance on our experiments. Hidden dimension size 100 is used for each word LSTM and 32 for each char LSTM, resulting in each turn representation of size 528 (utterance + response) + #features. Adam optimizer was used to minimize cross entropy loss, with a 1e-4 learning rate. At the fully connected layer, a dropout rate of 0.5 is used. These hyper-parameters were obtained after tuning them to our Alexa validation data, but can be easily tuned for different conversational tasks. Our PyTorch implementation and models are available for the research community[1].

[1] Available at *https://github.com/emory-irlab/ConvSAT*

Chapter 5

Experiments and Main Results

In this chapter, experimental setups such as obtaining online satisfaction labels and prediction tasks are discussed. Main results on dialogue breakdown detection, offline satisfaction prediction and online satisfaction prediction are presented next. This chapter concludes with heuristic performance analysis, feature importance study and representative error analysis. Some of the materials in this chapter was previously published in this reference [12].

5.1 Experimental Setting

In this section, I summarize the experimental settings, baseline methods and evaluation metrics.

5.1.1 Label generation for Alexa Prize dataset

Since online satisfaction annotation is extremely time-consuming, it is not feasible to generate all the necessary labels for training. Moreover, because of privacy issues with Amazon customers, I cannot outsource the annotation task to a public service like Amazon Mechanical Turk. Given the small size of human-labeled data, training on it is unrealistic. Based on these limitations, my proposed solution is to apply data programming to generate training data by using heuristic weak supervision strategies. I combine my domain heuristics to design a set of simple rule-based labeling functions [41, 43] to generate online training labels. Once large-scale training data is generated, the goal is to compare heuristic performance with proposed models to see if models can learn beyond these simple rules. The details of my labeling process are described below.

- Start introduction with 3.0 rating
- Label SAT for each engagement of depth $>=$ 4
- Label SAT for 4+ consecutive affirmation intents
- Label DSAT for 4+ consecutive negation intents
- Label DSAT for 4+ consecutive unidentified intents
- Final rating is from real users
- For remaining unlabeled turns, use continuous imputation

3.0 rating for the introduction was chosen under the assumption that every user initiates a conversation with medium-level satisfaction. The 4+ threshold for SAT and DSAT conditions were chosen since the average engagement depth on popular domains ranged between 2.0 to 3.5. Hence, any engagement that lasted longer than mean depth was considered successful. I included similar condition for consecutive affirmation and intents since our system frequently suggested relevant topics or entities to users. Hence, affirmation and negation intents can reflect user's attitude towards our system's suggestions. Unidentified intents were added since these receiving these intents were guaranteed to a failure or abrutpive transitions. Lastly, the rating of final turn was obtained from real user ratings, and remaining values were continuously imputed. The illustration of heuristic labeling is shown on Figure 5.1.

According to Figure 5.1, the turns that are affected by above heuristic labeling rules are the 1st turn, 4th turn and the last turn. The 1st turn (introduction) starts with 3.0 rating, and after 4 turns of engagement on travel domain, the 4th turn is labeled as 5.0 since it satisfies the second heuristic condition. Given the user rating is 3.0 for this conversation, all the unlabeled turns are continuously imputed based on these known (pivot) labels.

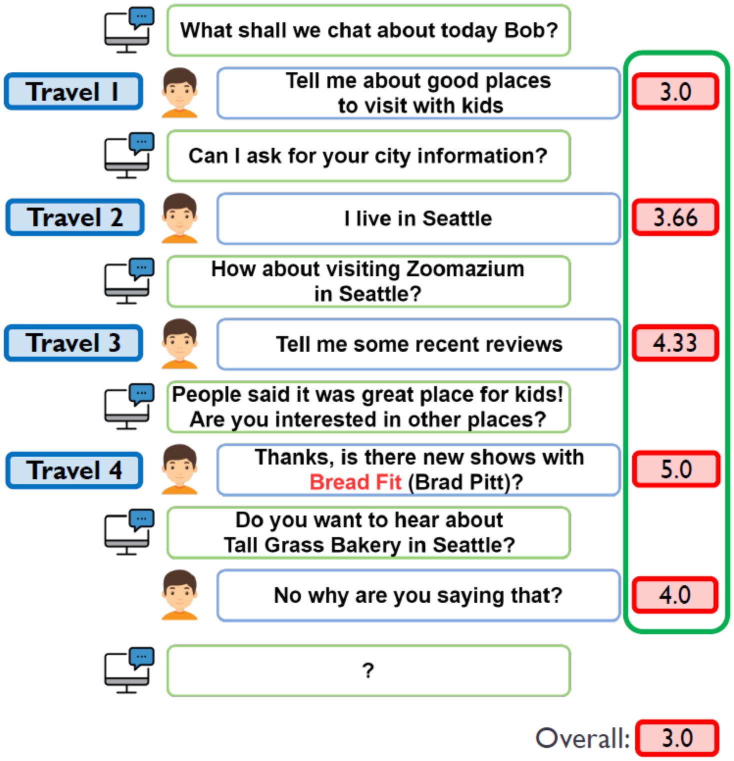

What shall we chat about today Bob?
Travel 1
Tell me about good places to visit with kids
3.0
Can I ask for your city information?
Travel 2
I live in Seattle
3.66
How about visiting Zoomazium in Seattle?
Travel 3
Tell me some recent reviews
4.33
People said it was great place for kids! Are you interested in other places?
Travel 4
Thanks, is there new shows with Bread Fit (Brad Pitt)?
5.0
Do you want to hear about Tall Grass Bakery in Seattle?
No why are you saying that?
4.0
?
Overall: 3.0

Figure 5.1: Illustration of heuristic labeling process for generating online training labels.

Since these labels are heuristically generated, I measured the statistical correlation of heuristic labeling to human annotated labels by applying these rules to my test data. The Fleiss Kappa score was **0.46**, which indicated moderate agreement. Hence, I hypothesize that these rules are reliable heuristics to generate large-scale training data. I emphasize that the heuristic labeling was done to generate *training data only*. The test data was manually annotated by two independent internal judges.

5.1.2 Baseline Methods

I define my first baseline method as a non-contextual bi-LSTM model (LSTM). This model only looks at the current utterance and response, which is equivalent of setting contextual window size W as 1. For state-of-the-art (SOTA) baseline, a contextual bi-LSTM (CLSTM), introduced by Hashemi et al. [24], models satisfaction based on intent-sensitive word embeddings. For DBDC3 data, I additionally report the best performing model (KTH Entry) participant on this challenge, which is a contextual LSTM model combined with a bag-of-words, averaged word embeddings, and handcrafted features [36]. Additionally, heuristic labeling (HL) baseline is reported for the online satisfaction task. Table 5.1 summarizes all the methods compared in my experiments.

5.1.3 Prediction Tasks

Based on the datasets defined in Chapter 3, I define three classification tasks: 1) dialogue breakdown detection; 2) online satisfaction prediction; 3) offline satisfaction prediction.

Dialogue Breakdown Detection Given a conversation turn (i), which is a concatenated vector of $[{U_i}^w;\ {R_i}^w;\ {U_i}^c;\ {R_i}^c;\ \hat{V}_i]$ defined in Section 4.1,

Model	**Name**
KTH Entry to DBDC3 challenge	KTH
Heuristic labeling for online Alexa dataset	HL
Non-contextual bi-LSTM	LSTM
Contextual bi-LSTM (SOTA)	CLSTM
Our method	ConvSAT

Table 5.1: Summary of methods compared.

predict the dialogue breakdown label ${B^i}_{pred}$ of each turn:

$${B^{\mathrm{i}}}_{pred} \in (NB, PB, B) \tag{5.1}$$

where *NB*, *PB*, and *B* represent "not breakdown", "possible breakdown" and "breakdown", respectively.

Online Satisfaction Prediction I define two states for the dialogue: $DSAT$ for dis-satisfied (equivalent to "breakdown") and SAT for satisfied (equivalent to "not breakdown"). Given each T_i, conditioned on previous turns, I predict the most likely binary satisfaction label ${S^i}_{pred}$ of each turn:

$$S^i_{pred} \in (SAT, DSAT) \tag{5.2}$$

Offline Satisfaction Prediction Given a session of length N turns, I predict S^N_{pred} at the end (T_N) of the conversation:

$$S^N_{pred} \in (SAT, DSAT) \tag{5.3}$$

Please note that at the last turn of the conversation, the online and offline prediction tasks are equivalent.

5.1.4 Evaluation Metrics and Training Details

For DBDC3 task, I stay consistent with the official evaluation metrics, which are micro-averaged accuracy and macro-averaged f1 on the breakdown label. I will additionally report precision and recall on breakdown labels for my implemented models. For the Alexa dataset, consistent with the DBDC3 setup, I report the micro-averaged accuracy and macro-averaged values of precision, recall, and f1 scores for both SAT and DSAT classes.

For DBDC3 data, since my behavioral features are designed for my Alexa Prize system, some of the features related to latency, ASR, and detailed topic-specific features are not available. Hence, these features are excluded when training on DBDC3 data. For word encoders, the hidden dimension was set to 64 to prevent overfitting. I used softmax activation on output layers for DBDC3 data (since it is a multi-class problem) and sigmoid activation for Alexa data (more appropriate for the binary classification problem). All the other settings, including the model architecture (described in Section 3.3) remained identical.

5.2 Main Results

In this section, dialogue breakdown & satisfaction prediction results, heuristic analysis, feature importance analysis and representative error study are presented.

5.2.1 Dialogue Breakdown Detection Results

ConvSAT significantly outperformed all the baseline models on accuracy, precision, recall and f1 for dialogue breakdown detection task, as shown in Table 5.2. There are 14.7% and 36.1% improvement in accuracy and f1 compared to KTH entry. Precision and recall for KTH entry are left blank because the official metrics did not include these. Similarly, ConvSAT improved the SOTA baseline by 2.4% on accuracy and 5.5% on f1 score, indicating statistically significant improvements with $p < 0.05$, measured by two-tailed Student's t-test. To ensure stability of the results and improvements, I report the mean and standard deviation of ConvSAT performance on five random test folds of 40 conversations each. Higher deviations in recall mostly occur between B and PB labels, indicating that the distinction between these two labels is the most challenging. Nonetheless, it is clear that leveraging sub-word information and behavioral feature matrices are beneficial for predicting failure.

Model	**AC**	**PR(B)**	**RC(B)**	**F1(B)**
KTH Entry	0.441	-	-	0.349
LSTM	0.456	0.322	0.566	0.410
CLSTM	0.494	0.351	0.625	0.450
ConvSAT	**0.506***±0.9	**0.374***±0.8	**0.651***±2.6	**0.475***±1.0
Impr. over KTH	14.7%	-	-	36.1%
Impr. over LSTM	10.9%	16.1%	15.0%	15.8%
Impr. over CLSTM	2.4%	6.5%	4.1%	5.5%

Table 5.2: Accuracy (AC), precision (PR), recall (RC) and f1 scores for dialogue breakdown detection. "B" stands for the breakdown label. "*" indicates statistical significance of improvement based on two-tailed Student's t-test with $p < 0.05$, compared to CLSTM.

We highlight that there is a significant gap in KTH entry and my re-implemented LSTM baseline (the LSTM baseline exhibits higher performance). The reason is due to a seemingly minor change in utterance representation. For KTH entry in the DBDC3 challenge, each utterance was represented by averaging the Google's Word2Vec embeddings with pre-trained vectors, while my implementation of the LSTM baseline considers each word separately. This is significant because averaging simplifies the training process but loses the temporal relationship between each word. Moreover, KTH entry represented each turn differently from my LSTM baseline by treating each utterance and response as separate timestamps. This doubles the length of the original sequence, and required insertion of dummy labels for each utterance to satisfy the length of predictions to be same as the input. During prediction, the *argmax* on three true labels were applied to each system response, ignoring the dummy label. In contrast, my LSTM baseline avoids this complexity by having two separate networks to represent each utterance and response separately. As a result, since my re-implementation of the baseline LSTM-based approach (inspired by the KTH entry) exhibits substantially higher performance on all metrics on this benchmark dataset, I use my LSTM implementation as the baseline for all subsequent Alexa experiments.

5.2.2 Online Satisfaction Prediction Results

ConvSAT improved all three baseline models on the online satisfaction prediction task, as reported in Table 5.3, with significant improvements over all the baselines on all metrics. This provides strong evidence that behavioral features and character information enable significant gains in real-world conversations. Compared to my heuristic baseline, ConvSAT showed 7.8% improvement in both accuracy and f1 respectively. Compared to the recent SOTA baseline, ConvSAT also improved by 2.4% and 2.2% on accuracy and f1 respectively, with all improvements significant with $p < 0.05$.

ConvSAT achieved 0.786 precision, 0.865 recall and 0.823 f1 for the DSAT label. For the SAT label, 0.804 precision, 0.701 recall and 0.749 f1 were achieved. The standard deviations are also computed based on random 5 test folds. This shows that predicting SAT label correctly is harder than

Model	**AC**	**PR**	**RC**	**F1**
HL	0.735	0.731	0.728	0.729
LSTM	0.749	0.763	0.732	0.734
CLSTM	0.774	0.772	0.767	0.769
ConvSAT	**0.793***±0.8	**0.795***±1.6	**0.783***±1.4	**0.786***±1.3
Impr. over HL	+7.8%	+8.7%	+7.5%	+7.8%
Impr. over LSTM	+5.8%	+4.1%	+6.9%	+7.0%
Impr. over CLSTM	+2.4%	+2.9%	+2.0%	+2.2%

Table 5.3: Online satisfaction prediction accuracy, precision, recall and f1 scores for detecting the SAT label in the Alexa Prize 2018 dataset.

correctly classifying DSAT label. Intuitively, satisfactory conditions should be more subjective than failure conditions because people can still dislike the conversation simply because the responses are boring or lack coherence. However, there are more explicit signals of failures, such as low ASR confidence, profane utterances and high latency.

5.2.3 Offline Satisfaction Prediction Results

For offline satisfaction prediction, I noticed that the general performance is lower compared to the online prediction results. This is because offline satisfaction prediction requires more complex reasoning that spans from the beginning to the end of conversations. Since my conversations have, on average, over 16 turns, I expect the decision boundaries to be more complex.

Nonetheless, ConvSAT outperforms the two state of the art baseline models significantly. There are 11.4%, 11.1% increases in accuracy and f1, respectively, compared to the non-contextual LSTM, and 3.1%, 3.4% boost compared to the contextual LSTM baseline. ConvSAT achieved 0.864 precision,

Model	**AC**	**PR**	**RC**	**F1**
LSTM	0.656	0.679	0.683	0.656
CLSTM	0.709	0.706	0.717	0.705
ConvSAT	**0.731***±2.1	**0.738***±0.7	**0.750***±1.0	**0.729***±2.0
Impr. over LSTM	11.4%	8.6%	9.8%	11.1%
Impr. over CLSTM	3.1%	4.5%	4.6%	3.4%

Table 5.4: Offline satisfaction prediction accuracy, precision, recall and f1 scores for detecting the SAT label in the Alexa Prize 2018 dataset.

0.667 recall and 0.752 f1 for DSAT. For SAT labels, ConvSAT achieved 0.612 precision, 0.833 recall, and 0.706 f1 score, which follows a similar pattern to online satisfaction results.

5.3 Discussion and Error Analysis

In this section, I first compared the performance between ConvSAT and heuristically generated labels to understand in which situations the model performed better than heuristics, and why the improvements were significant. Then, to understand the impact of different features groups, I conducted a feature ablation study on ConvSAT by systematically removing text representation and behavioral features. This section is concluded with a representative error analysis from dialogue breakdown predictions to illustrate open challenges.

5.3.1 Generalizing from Heuristic Labels

The online satisfaction results showed that all the baseline models including ConvSAT were able to learn from heuristically generated labels to predict more accurate labels. The most common mistake from heuristic labels was when conversations contained many short engagements. Since the heuristic explicitly used 4.0 threshold to identify satisfactory or unsatisfactory turns, these information were often ignored.

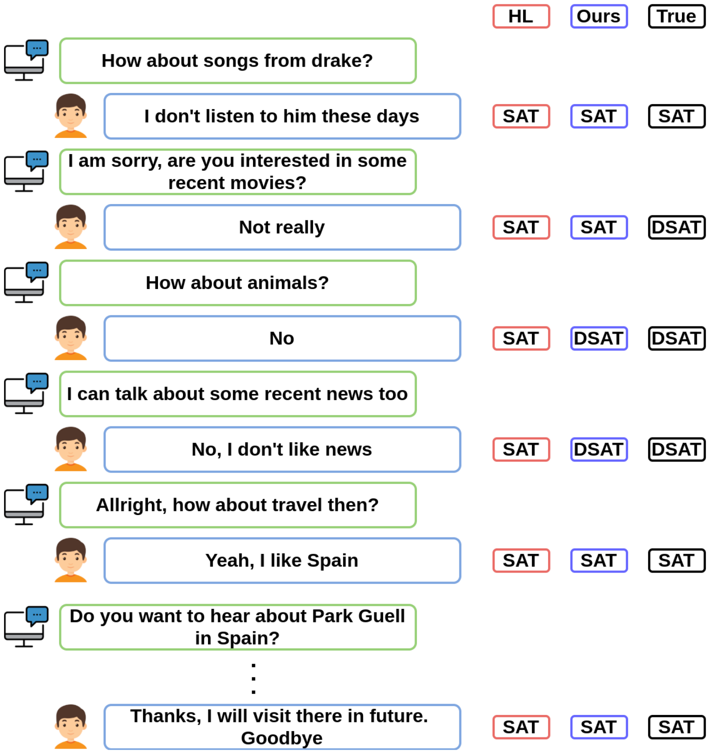
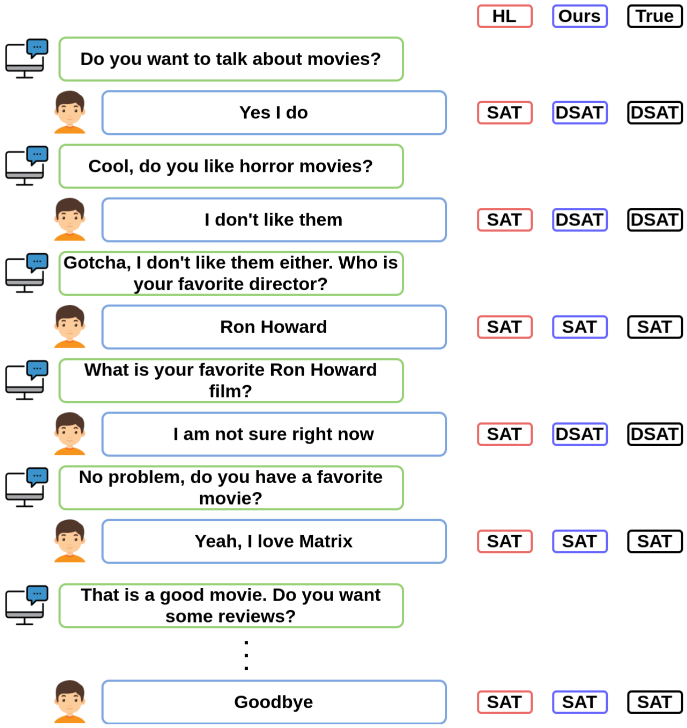

HL
Ours
True
How about songs from drake?
I don't listen to him these days
SAT SAT SAT
I am sorry, are you interested in some recent movies?
Not really
SAT SAT DSAT
How about animals?
No
SAT DSAT DSAT
I can talk about some recent news too
No, I don't like news
SAT DSAT DSAT
Allright, how about travel then?
Yeah, I like Spain
SAT SAT SAT
Do you want to hear about Park Guell in Spain?
Thanks, I will visit there in future. Goodbye
SAT SAT SAT

Figure 5.2: The first example conversation that replicated the original content to emphasize differences between heuristic labels and predicted labels from ConvSAT.

For instance, the turns from figure 5.2 shows a typical interaction when our system aggressively recommended other topics to regain user's interest. The first turn here was a terminal point from a successful 5+ turn engagement on music domain since our system failed to understand the utterance "I do not listen to him these days". At this point, the true label is still satisfactory since the user did not intend to switch topic. Both heuristic labels and predicted labels from ConvSAT correctly classified this turn since previous 4 turns (hidden in figure) all talked about drake.

However, for the next 3 turns, our system struggled to engage with users and received 3 consecutive negative intents. Based on the proposed labeling criteria, heuristic labeling is unable to capture this information since the observed turns are smaller than the threshold value (3<4). Since this conversation ended with another 4+ turn engagement on travel domain, heuristic labeling considered all the intermediate turns to be positive. I emphasize that if the intermediate turns had another negation intent, heuristic labels could have captured this information.

Despite some noise, ConvSAT successfully identified a decreasing trend in user satisfaction. Beginning from the 3rd turn, ConvSAT predicted DSAT labels until user engaged back to travel domain. During training, even though the model was fitted to replicate heuristic labels, feeding thousands of different conversations helped my model to learn a generalized pattern. For instance, a different user could have left low rating and ended the conversation when our system suggested travel domain. In this case, the missing values will be imputed to generate a decreasing trend and my model is exposed to a different scenario. As our model identified this trend, it verifies that there is a negative trend between the number of transition turns to user satisfaction.



	HL	Ours	True
Do you want to talk about movies?			
Yes I do	SAT	DSAT	DSAT
Cool, do you like horror movies?			
I don't like them	SAT	DSAT	DSAT
Gotcha, I don't like them either. Who is your favorite director?			
Ron Howard	SAT	SAT	SAT
What is your favorite Ron Howard film?			
I am not sure right now	SAT	DSAT	DSAT
No problem, do you have a favorite movie?			
Yeah, I love Matrix	SAT	SAT	SAT
That is a good movie. Do you want some reviews?			
⋮			
Goodbye	SAT	SAT	SAT

Figure 5.3: The second example conversation that replicated the original content to emphasize differences between heuristic labels and predicted labels from ConvSAT.

Here is another example conversation when ConvSAT outperformed heuristic labels, illustrated in Figure 5.3. In this case, our system was having a successful engagement in movies domain except that user did not like our suggestions. Since heuristic labeling ignored textual information but only relied on system states, all of the generated labels were positive. However, as my model was trained on extensive list of features, it was likely that many unsatisfactory turns contained common phrases (i.e. I am not sure). As a result, ConvSAT was not confident whether these turns were satisfactory or unsatisfactory, and returned a sequence of labels that were centered around the decision boundary (rating=3.5). I hypothesize that the remaining negative signals from utterances (i.e. sentiment scores) further contributed to correctly classify these turns.

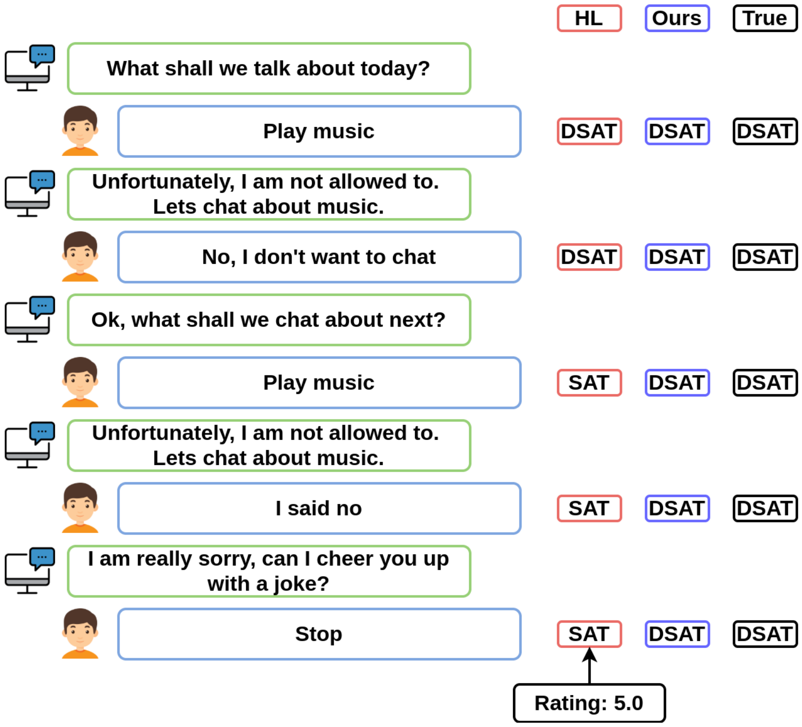

HL
Ours
True
What shall we talk about today?
Play music
DSAT
DSAT
DSAT
Unfortunately, I am not allowed to. Lets chat about music.
No, I don't want to chat
DSAT
DSAT
DSAT
Ok, what shall we chat about next?
Play music
SAT
DSAT
DSAT
Unfortunately, I am not allowed to. Lets chat about music.
I said no
SAT
DSAT
DSAT
I am really sorry, can I cheer you up with a joke?
Stop
SAT
DSAT
DSAT
Rating: 5.0

Figure 5.4: The third example conversation that replicated the original content to emphasize differences between heuristic labels and predicted labels from ConvSAT.

The last example (Figure 5.4) illustrates a scenario when heuristic labeling was misguided by the offline rating. In this example, Alexa users wanted to play a music through *Irisbot*, which was an unsupported feature for us but supported for commercial Alexa devices. Surprisingly, this user issued a 5.0 rating to this conversation despite the conversation quality looking terrible. I highlight that during label generation, all the offline labels were obtained from real user ratings since this was the only information available to training data. Since none of the criteria matched for these turns, all the labels were simply imputed from 3.0 (introduction) to the final rating. However, ConvSAT was very confident that all the turns in Figure 5.4 was negative since other similar conversations often ended with low ratings.

To conclude, the major improvement came from generalising to many different conversations. Open-domain conversations are very noisy, and it is very difficult to design a good heuristic to capture non-deterministic nature of conversations. First, some of these examples illustrated how small variations in system states and user utterances could avoid heuristic detection. Similarly, heuristic was misguided when there was small information available, or when user rating was highly subjective. Since my test dataset contained about 100 dialogues, these noises could have degraded the performance. Lastly, since ConvSAT had capability to jointly model textual evidence and behavioral feature matrix, I showed how ConvSAT could identify fluctuations in user satisfaction within single engagement, which is not supported by heuristic labels.

5.3.2 Feature Ablation

To show the effect of behavioral features and character information, I conducted an ablation study on both datasets by systematically removing these portions from ConvSAT. Table 5.5 shows the feature ablation results on online satisfaction and breakdown detection tasks. I used the same evaluation metrics defined for each task.

The results show that removing both behavioral features and character information decreases the accuracy and f1 on both datasets. In general, the decrease is much greater when removing behavioral features over removing characters. It shows that word-level information already contains most information, and in the future, more advanced subword representation such as phonetic representation needs to be explored.

Model	**AC(S)**	**F1(S)**	**AC(B)**	**F1(B)**	**BF**	**C**
ConvSAT (full)	0.793	0.786	0.506	0.475	✓	✓
- Characters	0.792	0.784	0.505	0.472	✓	✗
%Change	-0.1%	-0.2%	-0.2%	-0.6%	-	-
- Behavior	0.773	0.769	0.494	0.450	✗	✗
%Change	-2.5%	-2.1%	-2.3%	-5.2%	-	-

Table 5.5: Feature ablation on online satisfaction (S) and dialogue breakdown detection (B) tasks. “BF” and “C” stand for behavioral features and character features, respectively.

To conclude, distributional semantics are important features since they help models to learn the general context. However, I claim that they are not sufficient to model complex interactions between textual data and subjective satisfaction. For instance, a phrase I am done can be a strong signal of dissatisfaction after recent failures. However, after several successful engagements on multiple topics, the same phrase can represent a satisfaction or topic completion signal. Using distributional semantics alone, the model is likely to generalize on more frequent cases without learning the conversational flow effectively. Hence, I conjecture that my model successfully captures the behavioral features’ interaction with semantics, resulting in significant performance improvements over semantics alone.

5.3.3 Importance of Behavioral Features

Since I confirmed the importance of general behavioral signals, I now delve into specific behavioral feature importance. To understand the importance of each signal, I trained a gradient boosted decision tree (GBDT) by only using the behavioral feature matrices. I selected this tree-based model because of easy interpretability and support for categorical features. I used grid search to optimize the GBDT parameters, and used 5-fold cross validation to better generalize my model. Figure 5.5 reports the top 10 features learned for this task, using binary logistic loss function. I trained GBDT only on online Alexa data because I have a more comprehensive set of features, and substantially larger samples compared to the DBDC3 dataset.

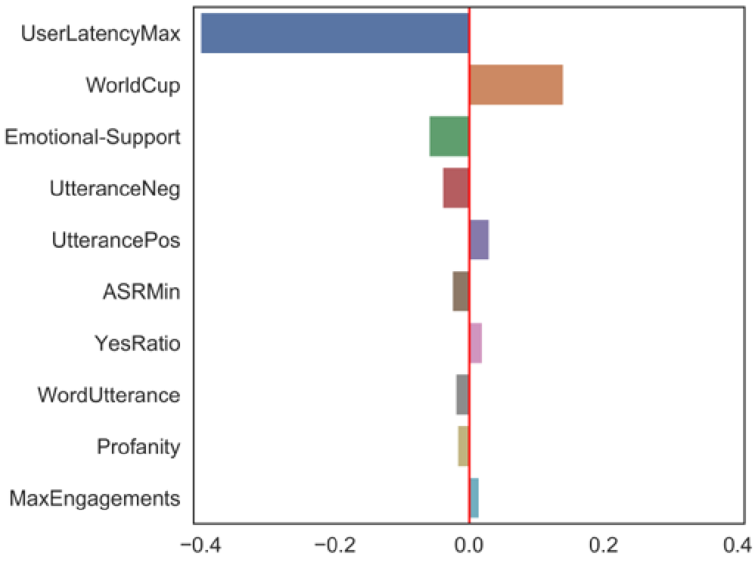


Figure 5.5: Top 10 feature importance from gradient boosted decision tree, trained on online satisfaction Alexa data. The left column indicates negatively contributing features while the right column indicates positively contributing features.

The results show that maximum user latency, UserLatencyMax (the amount of time a user spends before speaking) is a strong signal for online user (dis-)satisfaction. There are several potential reasons for high user latency, such as user not understanding the system's response, or simply does not know what to say after a nonsensical response. An increase in user latency also potentially compounds ASR errors, as the user's utterance is more likely to be cut off after a long pause.

Talking about the Worldcup was one of the most popular topics people enjoyed, while the emotional support topic was only invoked when user's mood seemed negative. These two features show that adding a topic distribution vector can help the model to learn the effect of an individual topic on user satisfaction. Sentiment features were also important because it is natural to expect a strong correlation between user sentiment and user satisfaction. Minimum ASR confidence value is another strong indicator of potential ASR failure and strongly contributed to importance. Lastly, utterance word counts, engagement (topic depth), and profanity (user cursing) are found to be strongly correlated with satisfaction and dis-satisfaction, respectively.

5.3.4 Representative Error Analysis

Since I cannot release the real Alexa utterances, my representative error analysis is performed only on the DBDC3 data. After comparing the prediction results between ConvSAT and the SOTA baseline, I present a representative case study where both models failed. The true label is indicated inside the bracket next to the final response.

BOT: Hello, I really like movies. How about we talk about movies?

USER: Sure. What's the most recent movie you've seen?

BOT: Honestly, you know, they're also different but we were talking before about the Wolf of Wall Street.

USER: I liked that movie a lot. I also liked inception.

BOT: Well, you know, I'm not sure. But anyway, I'd like to chat with you. [Not Breakdown]

For this case, the CLSTM baseline predicted "breakdown" and ConvSAT predicted "potential breakdown", while the true label was "not breakdown". First, it is likely that satisfaction prediction models can be biased to common phrases such as "I am not sure" or "I am sorry" since these phrases are frequent on unsatisfactory conversations. Understanding mixture of intents turned out to be challenging since both of the models did not understand user's secondary intent to continue conversation. Nonetheless, ConvSAT predicted the closer label to "not breakdown" and verifies the effectiveness of my proposed method.

Chapter 6

Applications of ConvSAT

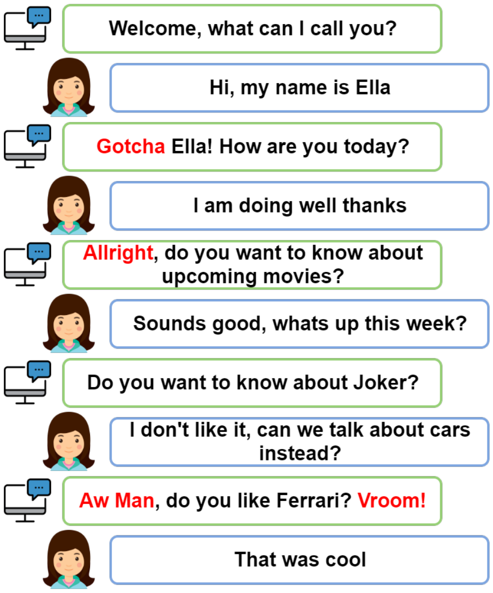
In this chapter, I present one application of how satisfaction prediction can be used as a proxy for evaluation. This application quantifies the change in user satisfaction after our team added prosody modulation feature to *Irisbot*. The materials in this chapter was previously published in this reference [11].

6.1 Quantifying Prosody Modulation Effects

Prosody modulation was added to our system responses to avoid monotonous and boring tones via commonly available Speech Synthesis Markup Language (SSML) [47]. For this experiment, our team replaced common phrases (i.e. filter words or interjections) with prerecorded Speechcons from Alexa Skills Kit APIs[1]. In some cases, the pitch and rate of these Speechcons are additioanlly tuned to convey excitement, hesitation and emphasis, allowing the agent to deliver a variety of empathetic responses to users. Intuitively, this approach should improve the quality of conversations, but attempts to *quantify* the effects of prosodic modulation on user satisfaction and engage-

[1] `https://https://developer.amazon.com/en-US/docs/alexa/custom-skills/speechcon-reference-interjections-english-us.html`

ment remain unclear. To accomplish this, I measured the effects of prosodic modulation on user behavior and engagement across multiple conversation domains, both immediately after tuned responses, and at the overall conversation level. The example conversation[2] provided in Figure 6.1 shows how my system utilized prerecorded Speechcons such as “Allright” or “Aw Man” to improve naturalness in conversations.



Welcome, what can I call you?

Hi, my name is Ella

Gotcha Ella! How are you today?

I am doing well thanks

Allright, do you want to know about upcoming movies?

Sounds good, whats up this week?

Do you want to know about Joker?

I don't like it, can we talk about cars instead?

Aw Man, do you like Ferrari? Vroom!

That was cool

Figure 6.1: Sample human-machine conversation from Irisbot. The red texts show response examples after inserting prerecorded Speechcons to convey artificial emotion.

6.1.1 Controlled Dataset Selection

The two versions selected for this study, A and B, are collected from July 23rd - July 27th and July 25th - July 31st. The only difference between these

[2] Due to the Alexa Prize data confidentiality rules, I cannot reproduce an actual user conversation, but the example represents a typical conversation with our system.

versions are the presence of prosody modulation feature. This controlled setup is to eliminate any potential change to different parts of the system that may affect the integrity of this evaluation. Please note that the overlap between these two periods is expected because our production server had 8 different instances for traffic control and A/B testing. Table 6.1 summarizes the statistics of two datasets A and B, each obtained from version A and B respectively.

	Dataset A	**Dataset B**
Prosody	✗	✓
Dialogues	1659	1202
Rated Dialogues	984 (59.3%)	670 (55.7%)
Average User Ratings	3.43	3.47
Average Turns	17.51	17.29

Table 6.1: Statistics on two datasets A and B, collected immediately before (A) and after (B) adding prosody modulation.

In general, both datasets have similar statistics. Even though dataset A has a slightly larger number of conversations than dataset B, the difference in averaged number of turns is small. The standard deviations of number of turns distributions are 14.75 and 14.36, indicating the diversity in conversation lengths for both datasets. Dataset A also has a slightly higher fraction of rated dialogues. After adding the prosody effect, there is a small increase of 0.04 in averaged user ratings. I emphasize that the only difference between these two datasets is the presence of prosody modification in system responses.

6.1.2 Proposed Metrics

We define engagements within conversations as sub-conversations that have 2 or more depth within the same domains. For instance, the conversation illustrated in Figure 6.1 has three distinct engagements, which are *opening* (depth=2), *movies* (depth=2) and *cars* (ongoing).

These domains are selected because they were the most popular, but most importantly, the earliest domains to utilize prosody modifications. Since other domains incorporated prosody modifications after version B, they were excluded from this study. I propose metrics in four different dimensions to measure user satisfaction (SAT): 1) immediate online satisfaction; 2) engagement-level satisfaction; 3) engagement depth; 4) user ratings.

First, I propose to capture the immediate effect on the predicted satisfaction after responses with prosody modifications, by computing the changes in the immediate satisfaction for the current turn ($\mathrm{SAT_i}$) and the next turn ($\mathrm{SAT_{i+1}}$). This is equivalent to measuring the difference in predicted satisfaction before and after the prosody modulation. These differences are summed and normalized by the count (N) of ($\mathrm{SAT_i}$, $\mathrm{SAT_{i+1}}$) pair per domain. I compute this metric as an immediate satisfaction difference ($SAT_{immediate}$):

$$SAT_{immediate} = \frac{\sum_1^N (SAT_{\mathrm{i+1}} - SAT_{\mathrm{i}})}{N} \tag{6.1}$$

We also compute the engagement-level difference in satisfaction ($SAT_{engagement}$) from the starting ($\mathrm{SAT_i}$) and ending ($\mathrm{SAT_{i+depth}}$) satisfaction of each engagement, with same normalization scheme where N is the total count of engagements per domain:

$$SAT_{engagement} = \frac{\sum_1^N (SAT_{\mathrm{i+depth}} - SAT_{\mathrm{i}})}{N} \tag{6.2}$$

Finally, I measure the differences in engagement *depth*, that is, the average number of turns a user spends conversing with each component. These three metrics are first computed on domain-specific level, and aggregated to measure the overall effect. Lastly, I report the averaged user satisfaction ratings (self-reported by Alexa users) to highlight the overall impact.

6.1.3 Pre-training Details

Because the goal is to measure exact changes in satisfaction across different turns, I trained ConvSAT in a regression setting to minimize the mean squared loss between predicted and annotated ratings. I used the generated training data and manually annotated test data from Chapter 3. Thus, I emphasize that even though the training labels were discrete, the model was trained to predict a continuous range of ratings. Regression fits much better to this task compared to binary classification setting because predicting continues range of values can better represent the magnitude of confidence compared to probability. Initially, heuristically generated labels scored **1.243** mean absolute error (MAE) on the test set. After training, the model achieved **0.772** MAE on the test set. Using this pre-trained model, all the turns in the two datasets are annotated with predicted satisfaction values.

6.1.4 Results and Discussion

According to the results reported in Table 6.2, the results are promising as they show improvements in all three metrics on diverse domains. When the results are aggregated for all six domains, there are 12.9%, 6.3% and 3.2% improvement on $SAT_{immediate}$, $SAT_{engagement}$ and depth, respectively. These improvements are statistically significant based on two-tailed Student's t-test (unpaired) with $p < 0.05$. The slight increase in user ratings from 3.43 to 3.47 further confirms that the improvements reflect the increased perceived quality in conversations.

Openings started with prosody modifications show improvements in all met-

Domains	$\boldsymbol{SAT_{immediate}}$	$\boldsymbol{SAT_{engagement}}$	**Depth**	**Samples**	**Prosody**
Opening	0.530	1.644	2.812	1514	✗
Movies	0.443	2.111	3.631	672	
Music	**0.454** (+8.8%)	1.535	3.506	569	
Games	0.380	1.685	3.666	573	
Travel	0.443	1.563	3.427	297	
News	0.413	1.274	3.555	378	
All	0.457	1.672	3.289	4003	
User ratings	3.43				
Opening	**0.536** (+1.1%)	**1.705** (+3.7%)*	**3.00** (+6.7%)*	1062	✓
Movies	**0.576** (+30.0%)*	**2.137** (+2.1%)	**3.790** (+4.4%)	377	
Music	0.414	**1.656** (+7.8%)*	**3.670** (+4.8%)	328	
Games	**0.499** (+31.3%)*	**1.718** (+1.9%)	**3.790** (+3.3%)	310	
Travel	**0.738** (+66.5%)*	**2.047** (+30.9%)*	**4.578** (+32.1%)*	19	
News	**0.426** (+3.1%)	**1.624** (+27.4%)*	**4.800** (+35.0%)*	25	
All	**0.516** (+12.9%)*	**1.778** (+6.3%)*	**3.395** (+3.2%)*	2121	
User ratings	**3.47** (+1.1%)				

Table 6.2: Change in online satisfaction difference ($SAT_{immediate}$), engagement-level satisfaction difference ($SAT_{engagement}$), conversation depth and averaged user ratings before (✗) and after (✓) adding prosody modification. "*" indicates statistical significance of improvement based on two-tailed Student's t-test with $p < 0.05$.

rics compared to the openings without prosody modifications. 1.1% increase in openings ($SAT_{immediate}$) is particularly interesting because I am measuring the change that is not conditioned to any previous context. I claim that the initial prosody modifications create a more positive first impression of our system, subsequently increasing $SAT_{engagement}$ and decreasing the likelihood to skip openings.

For each domain, *Travel* showed the strongest improvements on $SAT_{immediate}$

and $SAT_{engagement}$ metrics while *News* achieved the most increase in depth with statistical significance. One limitation is that the samples on these two domains are much less compared to other domains. *Movies* and *Games* domain, when evaluated on hundreds of samples, show that there are 30.0% and 31.3% statistically significant improvements on $SAT_{immediate}$. Depth and $SAT_{engagement}$ increased as well, but the changes are not statistically significant.

Surprisingly, for *Music* domain, there is a decrease in immediate satisfaction after prosody modifications. Unlike the *Travel* and *Games* components, where modified interjections occurred multiple times between engagements, *Music* conversations only modulated prosody rarely and not in a consistent way, indicating that prosody modulation must be carefully matched to the target domain, as I plan to explore in future work. In summary, my results showed that while overall both engagement and satisfaction increased when an agent becomes less monotonous and more "natural", the benefits vary across domains. For *Games*, *News*, and *Travel* domains the improvements are particularly noticeable, and less so for *Music* and *Movies* domains.

Chapter 7

Conclusions

This chapter concludes the thesis by providing the summary of the findings, main contributions and future research direction for this work.

7.1 Summary of the Results

First, I proposed a new satisfaction prediction model titled ConvSAT that combines multiple heterogeneous signals: 1) word-level representations; 2) char-level representations; 3) user behaviors; 4) topical preference; 5) system states and logs. ConvSAT is also unique that it supports both offline and online satisfaction prediction in a unified structure. I experimented with thousands of real open-domain conversations as well as publicly available DBDC3 dataset to conduct a large-scale study on predicting satisfaction and dialogue breakdown. The results are promising as ConvSAT outperformed state-of-the-art baselines in all three tasks, reaching 0.79 accuracy for the online satisfaction prediction task on Alexa Prize dataset.

These experiments demonstrate that aggregating aforementioned signals is needed when designing a successful satisfaction prediction model. In addition, I presented insights derived from feature ablation and importance for these tasks, showing that latency, topical, sentiment and ASR features are strong predictors of user (dis-)satisfaction.

Next, I used pretrained ConvSAT to quantify the effects of modulating prosody for conversational agent responses using our large scale, real-world dataset. Specifically, I confirmed that prosody modulation significantly effects immediate user satisfaction with an agent's responses, and that in some cases can also significantly increase the engagement of the users with the system, ultimately improving the overall subjective self-reported satisfaction ratings. While the overall improvements were significant, the effects were more dramatic in some domains, such as *Games* and *Travel*.

7.2 Contributions and Future Work

Conversational agents are being used widely in information-search, online bookings, and almost any setting where a human interaction could be valuable. While much prior work focused on the implementation and science behind these agents, this thesis focuses on developing new, automated ways to evaluate conversational agents in online using contextual, behavioral and system-specific signals. The predicted satisfaction could be used for both offline evaluation for improving conversational systems, or as online feedback for various downstream tasks. For commercial purposes, online satisfaction can be used for live monitoring of unexpected failures or a tool to understand user experience.

In future, it will be interesting to apply reinforcement learning techniques to use online satisfaction as rewards to learn a more sophisticated dialogue policy, or even a correction policy to revise responses in real-time. Moreover, since our character encoders did not contribute much to the gains, phonetic embeddings can be experimented to improve generalization on ASR errors. These future research directions can be milestones to enable a new generation of more responsive and intelligent conversational agents.

7.3 Acknowledgment

I gratefully acknowledge the computational and financial support from Amazon Alexa Prize. In addition, I acknowledge the computational and financial support from Home Depot Search Science team for supporting my research.

Bibliography

[1] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

[2] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proc. of SIGIR*, pages 19–26. ACM, 2006.

[3] Ali Ahmadvand, Ingyu Jason Choi, Harshita Sahijwani, Justus Schmidt, Mingyang Sun, Sergey Volokhin, Zihao Wang, and Eugene Agichtein. Emory irisbot: An open-domain conversational bot for personalized information access. *Alexa Prize Proceedings*, 2018.

[4] Ali Ahmadvand, Harshita Sahijwani, Ingyu Jason Choi, and Eugene Agichtein. ConCET: Entity-aware topic classification for open-domain conversational agents. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* ACM, 2019.

[5] James F Allen, Lenhart K Schubert, George Ferguson, Peter Heeman, Chung Hee Hwang, Tsuneaki Kato, Marc Light, Nathaniel Martin, Bradford Miller, Massimo Poesio, et al. The trains project: A case study in

building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(1):7–48, 1995.

[6] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[8] Dan Bohus and Alexander I Rudnicky. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[9] Chun-Yen Chen, Dian Yu, Weiming Wen, Yi Mang Yang, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, et al. Gunrock: Building a human-like social bot by leveraging large scale real user data. *Alexa Prize Proceedings*, 2018.

[10] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[11] Jason Ingyu Choi and Eugene Agichtein. Quantifying the effects of prosody modulation on user engagement and satisfaction in conversational systems. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, pages 417–421, 2020.

[12] Jason Ingyu Choi, Ali Ahmadvand, and Eugene Agichtein. Offline and online satisfaction prediction in open-domain conversational systems. In

Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pages 1281–1290, 2019.

[13] Kenneth Mark Colby. *Artificial paranoia: A computer simulation of paranoid processes*, volume 49. Elsevier, 2013.

[14] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

[15] William H DeLone and Ephraim R McLean. Information systems success: The quest for the dependent variable. *Information systems research*, 3(1):60–95, 1992.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[17] Hao Fang, Hao Cheng, Elizabeth Clark, Ariel Holtzman, Maarten Sap, Mari Ostendorf, Yejin Choi, and Noah A Smith. Sounding board–university of washington's alexa prize submission. *Alexa Prize Proceedings*, 2017.

[18] Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): a randomized controlled trial. *JMIR mental health*, 4(2):e19, 2017.

[19] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168, 2005.

[20] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] CJ Hutto Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proc. of ICWSM*, 2014.

[22] Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. Topic-based evaluation for conversational bots. *arXiv preprint arXiv:1801.03622*, 2018.

[23] Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*, 2019.

[24] Seyyed Hadi Hashemi, Kyle Williams, Ahmed El Kholy, Imed Zitouni, and Paul A Crook. Measuring user satisfaction on smart speaker intelligent assistants using intent sensitive query embeddings. In *Proc. of CIKM*, pages 1183–1192. ACM, 2018.

[25] Ryuichiro Higashinaka, Kotaro Funakoshi, Michimasa Inaba, Yuiko Tsunomori, Tetsuro Takahashi, and Nobuhiro Kaji. Overview of dialogue breakdown detection challenge 3. *Proceedings of Dialog System Technology Challenge*, 6, 2017.

[26] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[27] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences.

In *Advances in neural information processing systems*, pages 2042–2050, 2014.

[28] Zongcheng Ji, Zhengdong Lu, and Hang Li. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988*, 2014.

[29] Aishwarya Kamath and Rajarshi Das. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*, 2018.

[30] Chandra Khatri, Behnam Hedayatnia, Anu Venkatesh, Jeff Nunn, Yi Pan, Qing Liu, Han Song, Anna Gottardi, Sanjeev Kwatra, Sanju Pancholi, et al. Advancing the state of the art in open domain dialog systems through the alexa prize. *arXiv preprint arXiv:1812.10757*, 2018.

[31] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[32] Julia Kiseleva, Kyle Williams, Ahmed Hassan Awadallah, Aidan C Crook, Imed Zitouni, and Tasos Anastasakos. In *Proc. of SIGIR*, pages 45–54. ACM, 2016.

[33] Julia Kiseleva, Kyle Williams, Jiepu Jiang, Ahmed Hassan Awadallah, Aidan C Crook, Imed Zitouni, and Tasos Anastasakos. Understanding user satisfaction with intelligent assistants. In *Proc. of CHIIR*, pages 121–130. ACM, 2016.

[34] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[35] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[36] José Lopes. How generic can dialogue breakdown detection be? the kth entry to dbdc3. In *Proc. Dialog System Technology Challenges Workshop (DSTC6)*, 2017.

[37] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. of NeurIPS*, pages 3111–3119, 2013.

[38] Robert R Morris, Kareem Kouddous, Rohan Kshirsagar, and Stephen M Schueller. Towards an artificially empathic conversational agent for mental health applications: system design and user perceptions. *Journal of medical Internet research*, 20(6):e10148, 2018.

[39] Jan Pichl, Petr Marek, Jakub Konrád, Martin Matulík, Hoang Long Nguyen, and Jan Šedivỳ. Alquist: The alexa prize socialbot. *Alexa Prize Proceedings*, 2018.

[40] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*, 2018.

[41] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in neural information processing systems*, pages 3567–3575, 2016.

[42] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020.

[43] Tommy Sandbank, Michal Shmueli-Scheuer, Jonathan Herzig, David Konopnicki, John Richards, and David Piorkowski. Detecting egregious conversations between customers and virtual agents. *arXiv preprint arXiv:1711.05780*, 2017.

[44] Shumpei Sano, Nobuhiro Kaji, and Manabu Sassano. Predicting causes of reformulation in intelligent assistants. *arXiv preprint arXiv:1707.03968*, 2017.

[45] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015.

[46] Stefan Steidl, Christian Hacker, Christine Ruff, Anton Batliner, Elmar Nöth, and Jürgen Haas. Looking at the last two turns, i'd say this dialogue is doomed–measuring dialogue success. In *International Conference on Text, Speech and Dialogue*, pages 629–636. Springer, 2004.

[47] Paul Taylor and Amy Isard. Ssml: A speech synthesis markup language. *Speech communication*, 21(1-2):123–133, 1997.

[48] Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, et al. On evaluating and comparing open domain dialog systems. *arXiv preprint arXiv:1801.03625*, 2018.

[49] Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363, 2005.

[50] Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. Paradise: A framework for evaluating spoken dialogue agents. In *Proc. of ACL*, pages 271–280. Association for Computational Linguistics, 1997.

[51] Marilyn A Walker, Rebecca Passonneau, and Julie E Boland. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Proc. of ACL*, pages 515–522. Association for Computational Linguistics, 2001.

[52] Richard S Wallace. The anatomy of alice. In *Parsing the Turing Test*, pages 181–210. Springer, 2009.

[53] Heyuan Wang, Ziyi Wu, and Junyu Chen. Multi-turn response selection in retrieval-based chatbots with iterated attentive convolution matching network. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1081–1090, 2019.

[54] Zihao Wang, Ali Ahmadvand, Jason Ingyu Choi, Payam Karisani, and Eugene Agichtein. Emersonbot: Information-focused conversational ai emory university at the alexa prize 2017 challenge.

[55] Wei Wei, Jiayi Liu, Xianling Mao, Guibing Guo, Feida Zhu, Pan Zhou, and Yuchong Hu. Emotion-aware chat machine: Automatic emotional response generation for human-like emotional interaction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1401–1410, 2019.

[56] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[57] Kyle Williams, Julia Kiseleva, Aidan C Crook, Imed Zitouni, Ahmed Hassan Awadallah, and Madian Khabsa. Detecting good abandonment in mobile search. In *Proc. of WWW*, pages 495–505. IWWWSC, 2016.

[58] Barbara H Wixom and Peter A Todd. A theoretical integration of user satisfaction and technology acceptance. *Information systems research*, 16(1):85–102, 2005.

[59] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. *arXiv preprint arXiv:1612.01627*, 2016.

[60] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64, 2017.

[61] Rui Yan, Yiping Song, and Hua Wu. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64, 2016.

[62] Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. A hybrid retrieval-generation neural conversation model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1341–1350, 2019.

[63] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, Jun Huang, and Haiqing Chen. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 245–254, 2018.

[64] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93, 2020.