End-to-End Neural Modeling of Multi-Turn Technical Conversations at Scale

Tony Livins

Solent University

Abstract

This work investigates neural dialogue models trained in an end-to-end fashion using an expanded version of the Ubuntu Dialogue Corpus, a large-scale dataset containing close to one million multi-turn conversations, more than seven million utterances, and roughly one hundred million tokens. The corpus is notable for its size, extended contextual dependencies, and technical focus, making it suitable for training high-capacity models directly from data with minimal manual feature design. We establish benchmark results under two experimental settings. In the first, models are trained to identify the correct next response from a set of candidate utterances. In the second, models are trained to generate a response by maximizing the conditional log-likelihood given the dialogue context. Both settings are evaluated using a recall-based task referred to as next utterance classification, as well as embedding-based metrics that assess the topical relevance of responses. Our analysis shows that existing end-to-end approaches are unable to fully address these challenges. To better understand their limitations, we conduct a qualitative error analysis of classification failures and examine example outputs produced by generative models. Based on these findings, we outline several promising research directions for further work on the Ubuntu Dialogue Corpus and on end-to-end dialogue modeling more broadly.

1. Introduction

Developing statistical models that can engage in coherent and natural conversation with humans remains a central challenge in artificial intelligence. Historically, dialogue systems relied heavily on manually engineered features and carefully designed pipelines, which limited both the diversity of responses and the range of environments in which such systems could be deployed. Recent progress in neural language modeling has made it increasingly feasible to learn dialogue behavior directly from conversational data, with human involvement largely restricted to selecting model architectures and hyperparameters. Despite these advances, substantial obstacles remain before such systems can be applied reliably in real-world settings.

In this work, we focus on constructing dialogue agents using an end-to-end learning paradigm. In contrast to traditional modular approaches, end-to-end systems are trained directly from conversational data to optimize a single objective function. We base our study on the Ubuntu Dialogue Corpus, a large collection of nearly one million dyadic conversations extracted from Ubuntu technical support chat logs. These dialogues are multi-turn and unstructured, lacking an explicit symbolic representation of dialogue state

or intent. This distinguishes them from many prior datasets that emphasize structured interactions, such as slot-filling tasks commonly used in task-oriented dialogue research.

The motivation for building such a large, unstructured dialogue resource is grounded in broader trends across artificial intelligence research. In particular, progress in machine learning has been driven by three key factors: the public availability of large and diverse datasets, access to substantial computational resources, and the development of effective training methods for deep neural architectures, including techniques that leverage unlabeled data. Against this backdrop, the Ubuntu Dialogue Corpus provides a valuable testbed for investigating whether end-to-end neural models can learn meaningful conversational behavior directly from raw dialogue data.

1.1 Motivation for End-to-End Dialogue Systems

Dialogue systems have traditionally been designed as modular pipelines, where individual components such as language understanding, dialogue management, and response generation are trained separately, often with different objective functions. These systems typically operate over structured representations, such as predefined dialogue acts or slot-value pairs, which simplifies training and allows for explicit control over system behavior. As a result, modular approaches have achieved strong performance in narrowly defined, goal-oriented domains.

End-to-end dialogue systems depart from this paradigm by learning all internal representations directly from data, without relying on manually specified intermediate structures. In such systems, the entire model is optimized using a single training objective, regardless of whether the final output is produced through retrieval or generation. This unified training approach allows the model to automatically discover representations that are well suited to the data and task at hand.

One key advantage of end-to-end learning is flexibility. Once an architecture is defined, adapting the system to a new domain primarily requires additional training data rather than extensive feature engineering or redesign of internal components. This property is particularly appealing for open-domain or general-purpose dialogue systems, where it is impractical to predefine all possible states, actions, or user intents.

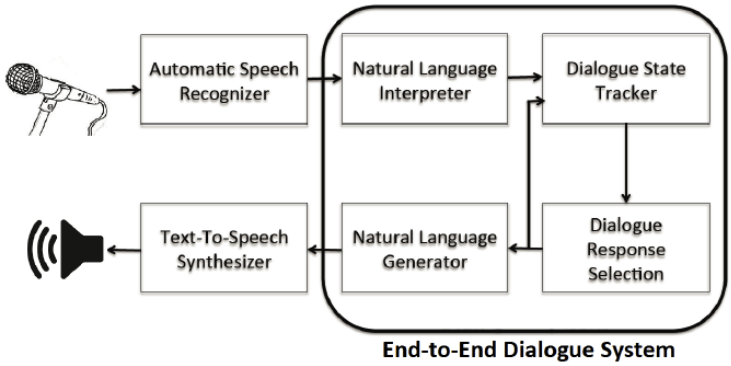

Figure 1 An end-to-end dialogue system substitutes the conventional modular components with one unified statistical model.

However, it remains unclear whether end-to-end methods can consistently outperform modular systems across a wide range of dialogue scenarios. While modular systems benefit from interpretability and explicit control, they often struggle to generalize beyond the domains for which they were designed. End-to-end approaches, on the other hand, offer the promise of scalability and adaptability, but their effectiveness depends heavily on the availability of large, high-quality conversational datasets. This uncertainty motivates a systematic examination of end-to-end dialogue models using a resource as large and complex as the Ubuntu Dialogue Corpus.

1.2 Paper Outline

The remainder of the paper is organized as follows. Section 2 reviews related work, including existing dialogue datasets and learning architectures for both structured and unstructured conversational settings. Section 3 describes the Ubuntu Dialogue Corpus in detail, covering corpus statistics, data collection procedures, and updates introduced in the current version. In Section 4, we present and evaluate response ranking models trained on the corpus, focusing on the task of next utterance classification and providing a qualitative analysis of common classification errors. Section 5 examines response generation models, assessing their performance using embedding-based similarity measures and illustrative examples. Finally, Section 6 concludes the paper by discussing limitations of the corpus and evaluation methods, and by outlining directions for future research in end-to-end dialogue systems.

2. Related Work

Research on dialogue systems spans a wide range of datasets and modeling approaches, reflecting differing assumptions about dialogue structure, supervision, and system

objectives. This section reviews prior work most relevant to end-to-end dialogue modeling, with a focus on existing dialogue datasets and neural learning architectures designed for conversational tasks.

2.1 Dialogue Datasets

Early dialogue research relied primarily on small, carefully curated datasets collected for specific applications, such as task-oriented spoken dialogue systems. These datasets often include explicit annotations for dialogue acts, slot values, and user intentions, enabling supervised training of modular system components. While such resources are valuable for controlled experiments, their limited size and narrow scope restrict the development of models capable of handling open-ended conversation.

More recent efforts have emphasized the construction of large-scale dialogue corpora extracted from real-world interactions. Examples include customer support logs, online chat transcripts, and social media conversations. These datasets typically contain millions of utterances and capture a wide range of linguistic phenomena, but they are largely unstructured and lack explicit semantic annotations. As a result, they present significant challenges for modeling, particularly with respect to long-term context and conversational coherence.

Several publicly available corpora have been introduced to support research in open-domain and multi-turn dialogue. These include datasets derived from online forums, microblogging platforms, and question-answering communities. Compared with these resources, the Ubuntu Dialogue Corpus is distinguished by its size, technical focus, and extended conversational contexts. The dialogues are centered around problem-solving interactions related to the Ubuntu operating system, which encourages coherent multi-turn exchanges while avoiding the heavy annotation requirements of traditional task-oriented datasets.

2.2 Learning Architectures for End-to-End Dialogue Systems

Early learning-based dialogue systems commonly adopted modular architectures, with separate models for language understanding, dialogue state tracking, policy learning, and response generation. These components were often trained independently using task-specific objectives, such as classification accuracy or reward maximization. While effective in constrained domains, such architectures require substantial manual design and do not scale easily to complex or open-ended conversational settings.

End-to-end learning approaches aim to overcome these limitations by training a single model to map dialogue context directly to system responses. In these frameworks, all intermediate representations are learned automatically from data, without relying on predefined dialogue states or handcrafted features. Neural network models, particularly those based on recurrent and attention-based architectures, have become the dominant choice for implementing end-to-end dialogue systems.

Two broad classes of end-to-end models have emerged. The first focuses on response selection, where the system chooses an appropriate reply from a fixed set of candidate utterances. These models are often trained using ranking or classification objectives and can leverage large datasets efficiently. The second class addresses response generation, where the model produces a new utterance token by token conditioned on the dialogue history. Generative models offer greater flexibility but are more difficult to train and evaluate.

Recent work has explored a variety of neural architectures for these tasks, including recurrent neural networks, hierarchical models that explicitly represent dialogue structure, and memory-augmented networks designed to capture long-term dependencies. Despite notable progress, existing approaches continue to struggle with maintaining coherence over long conversations and generating informative, contextually appropriate responses. These challenges motivate further investigation into both dataset design and model architecture for end-to-end dialogue systems.

3. The Ubuntu Dialogue Corpus

The Ubuntu Dialogue Corpus is a large-scale collection of multi-turn conversations drawn from technical support discussions related to the Ubuntu operating system. It is designed to support research on end-to-end dialogue modeling by providing long, coherent interactions without relying on manually annotated dialogue states or intents. The corpus captures natural problem-solving conversations and presents substantial challenges related to context tracking, response relevance, and long-term coherence.

3.1 Ubuntu Chat Logs

The source data for the corpus consists of chat logs collected from the official Ubuntu support channels hosted on Internet Relay Chat. These channels are used by individuals seeking help with Ubuntu-related issues, as well as by volunteers and experienced users who provide assistance. Conversations are informal, multi-participant, and span a wide range of technical topics, including system configuration, software installation, and troubleshooting.

The chat logs are timestamped and include speaker identifiers, but they do not contain explicit annotations indicating conversational structure or intended recipients. Messages may overlap in time, and multiple conversations often occur simultaneously within the same channel. As a result, transforming raw chat logs into coherent two-party dialogues requires additional processing.

Time	User	Utterance
03:44	Old	I dont run graphical ubuntu, I run ubuntu server.
03:45	kuja	Taru: Haha sucker.
03:45	Taru	Kuja: ?
03:45	bur[n]er	Old: you can use "ps ax" and "kill (PID#)"
03:45	kuja	Taru: Anyways, you made the changes right?
03:45	Taru	Kuja: Yes.
03:45	LiveCD	or killall speedlink
03:45	kuja	Taru: Then from the terminal type: sudo apt-get update
03:46	_pm	if i install the beta version, how can i update it when the final version comes out?
03:46	Taru	Kuja: I did.

Sender	Recipient	Utterance
Old		I dont run graphical ubuntu, I run ubuntu server.
bur[n]er	Old	you can use "ps ax" and "kill (PID#)"
kuja	Taru	Haha sucker.
Taru	Kuja	?
kuja	Taru	Anyways, you made the changes right?
Taru	Kuja	Yes.
kuja	Taru	Then from the terminal type: sudo apt-get update
Taru	Kuja	I did.

Figure 2 A sample chat room exchange from the #ubuntu channel in the Ubuntu Chat Logs is shown on the left, alongside the separated (disentangled) conversations used in the Ubuntu Dialogue Corpus on the right.

Time	User	Utterance
[12:21]	dell	well, can I move the drives?
[12:21]	cucho	dell: ah not like that
[12:21]	RC	dell: you can't move the drives
[12:21]	RC	dell: definitely not
[12:21]	dell	ok
[12:21]	dell	lol
[12:21]	RC	this is the problem with RAID:)
[12:21]	dell	RC haha yeah
[12:22]	dell	cucho, I guess I could just get an enclosure and copy via USB...
[12:22]	cucho	dell: i would advise you to get the disk

Sender	Recipient	Utterance
dell		well, can I move the drives?
cucho	dell	ah not like that
dell	cucho	I guess I could just get an enclosure and copy via USB
cucho	dell	i would advise you to get the disk
dell		well, can I move the drives?
RC	dell	you can't move the drives. definitely not. this is the problem with RAID :)
dell	RC	haha yeah

Figure 3 An illustration of the dialogue extraction process before (left) and after (right) the algorithm appends and merges utterances. Because RC interacts only with dell, all of RC's utterances are included. In contrast, this is not applied to dell, since he

3.2 Dataset Creation

Constructing the Ubuntu Dialogue Corpus from raw chat logs involves several preprocessing steps designed to extract dyadic conversations suitable for training dialogue models. The goal is to identify sequences of utterances exchanged between two participants that form a coherent interaction.

The dataset creation pipeline operates by grouping messages into conversations based on temporal proximity, speaker interaction patterns, and inferred addressee information. Each resulting dialogue consists of an ordered sequence of utterances, preserving the original message content while removing extraneous metadata.

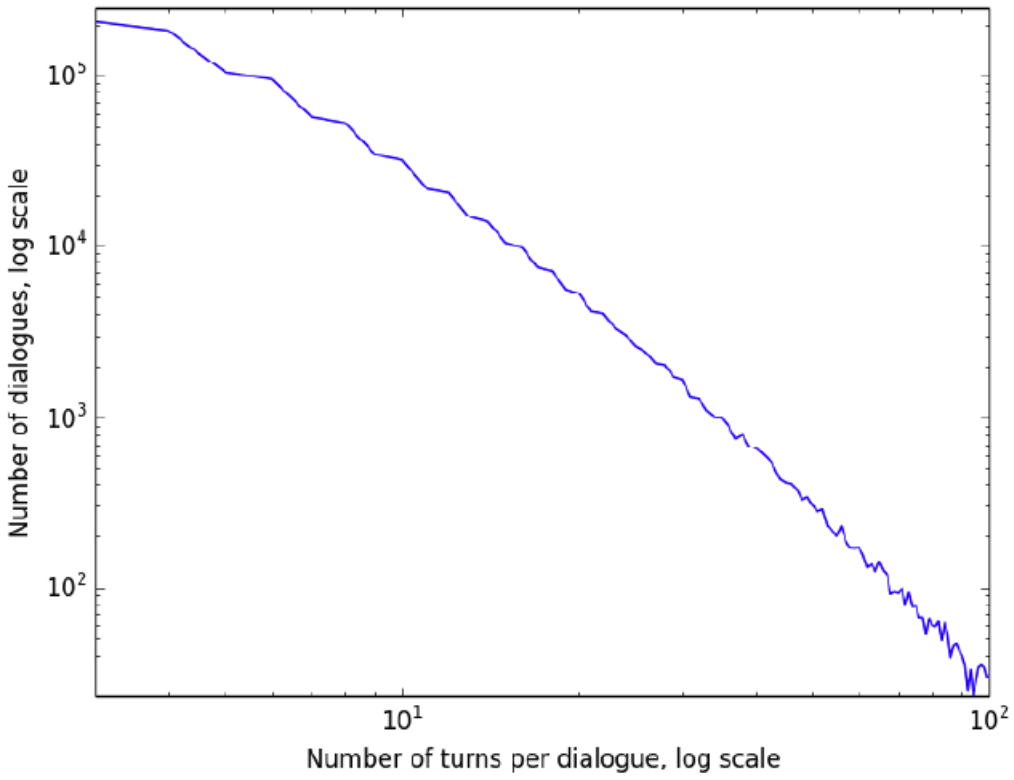

Figure 4 A plot showing the number of conversations for each turn count, with both axes displayed on a logarithmic scale.

3.2.1 Recipient Identification

A key challenge in processing multi-user chat logs is determining the intended recipient of each message. Since messages are posted in a shared channel, users frequently address specific individuals using explicit name mentions or conversational cues. The corpus construction process uses heuristic rules to infer the most likely recipient of each utterance.

These heuristics consider factors such as direct name references, recent interaction history, and message timing. When a clear recipient cannot be identified, the utterance may be excluded from the dataset to avoid introducing noise. While this approach does not guarantee perfect accuracy, it substantially improves the coherence of extracted dialogues.

3.2.2 Utterance Creation

Once recipient relationships are inferred, messages are grouped into utterances that form the building blocks of each dialogue. Consecutive messages sent by the same speaker within a short time window are merged into a single utterance, reflecting the natural tendency for users to split thoughts across multiple chat messages.

This aggregation step reduces fragmentation and produces utterances that more closely resemble turns in spoken or written conversation. The resulting utterances are tokenized and normalized to prepare them for use in training neural dialogue models.

# dialogues (human-human)	936,000
# utterances (in total)	7,100,000
# words (in total)	100,000,000
Min. # turns per dialogue	3
Avg. # turns per dialogue	7.71
Avg. # words per utterance	10.34
Median conversation length (min)	6
Training set dialogues	898,000
Validation/test set dialogues	19,000
Training set examples	unspecified

Figure 5 Characteristics of the Ubuntu Dialogue Corpus. It is important to note that the number of training examples can be set during the creation of the training set.

3.2.3 Special Cases and Limitations

Certain patterns in the chat logs present difficulties for dialogue extraction. These include messages addressed to multiple users, broadcast announcements, automated system messages, and off-topic chatter. The preprocessing pipeline applies filtering rules to handle such cases, either by removing problematic messages or by excluding entire interactions that fail to meet coherence criteria.

Despite these efforts, the resulting dataset is not free from noise. Errors in recipient identification and dialogue segmentation can occur, particularly in highly active channels.

These limitations reflect the inherent complexity of extracting structured conversations from unstructured, multi-party communication.

3.3 Dataset Statistics

The finalized Ubuntu Dialogue Corpus contains close to one million dialogues, comprising several million utterances and a very large number of tokens. Dialogues vary widely in length, with some consisting of only a few turns and others extending across dozens of utterances. This variability provides a realistic testbed for evaluating a model's ability to manage both short interactions and extended conversational contexts.

The vocabulary size is large due to the technical nature of the discussions, which include command-line instructions, file paths, error messages, and domain-specific terminology. These characteristics make the corpus particularly challenging for language models and highlight the need for architectures capable of handling rare words and long-range dependencies.

3.4 Test Set Generation

To support fair and consistent evaluation, a dedicated test set is constructed separately from the training and validation data. Dialogues included in the test set are selected to ensure that they do not overlap with those used for training, either directly or through shared conversational segments.

For evaluation tasks such as response ranking and generation, test examples are created by pairing dialogue contexts with candidate responses. These candidates include the true next utterance as well as distractor responses sampled from other dialogues. This setup allows models to be assessed on their ability to identify or generate responses that are both contextually appropriate and semantically relevant.

Context	Response	Flag
well, can I move the drives? _eot_ ah not like that	I guess I could just get an enclosure and copy via USB	1
well, can I move the drives? _eot_ ah not like that	you can use "ps ax" and "kill (PID #)"	0

Figure 6 An example from the test set in the format (context, reply, flag). The 'eot' tag indicates the end of a user's turn within the context, while the 'eou' tag marks the end of a user utterance without a change in turn.

4. Response Classification Architectures

This section describes the models used for response selection, where the task is to choose the most appropriate next utterance from a fixed set of candidates given a dialogue context. Each approach differs in how the dialogue history and candidate responses are represented and compared.

4.1 TF-IDF

The TF-IDF baseline represents dialogue contexts and candidate responses as sparse vectors weighted by term frequency and inverse document frequency. Similarity between a context and a candidate response is computed using cosine similarity. The response with the highest similarity score is selected as the prediction.

This approach does not incorporate word order or semantic relationships beyond shared vocabulary. Despite its simplicity, it provides a useful non-neural baseline that highlights the contribution of learned representations in more advanced models.

4.2 RNN Dual Encoder

The RNN Dual Encoder model uses two recurrent neural networks with shared parameters to encode the dialogue context and the candidate response independently. Each input sequence is processed token by token, and the final hidden state is taken as a fixed-length vector representation.

The similarity between the context representation and the response representation is computed using a learned bilinear transformation. The resulting score reflects how well the

candidate response matches the dialogue history. The model is trained to assign higher scores to correct responses than to incorrect ones using a discriminative objective.

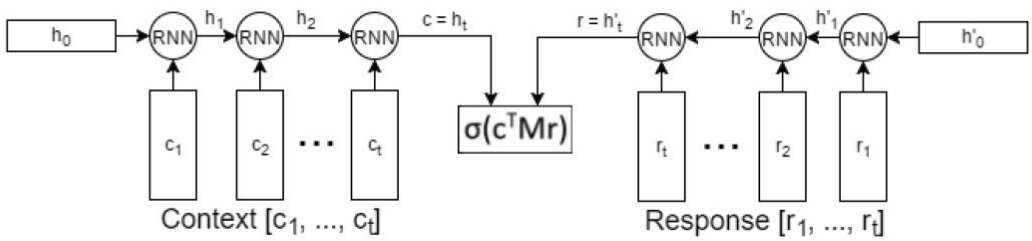

Figure 7 Diagram of the Dual Encoder (DE) model with shared RNN weights.

4.3 LSTM Dual Encoder

The LSTM Dual Encoder follows the same overall structure as the RNN Dual Encoder but replaces simple recurrent units with Long Short-Term Memory cells. This modification allows the model to better capture long-range dependencies within both the dialogue context and the response sequence.

By leveraging gating mechanisms, the LSTM-based encoder mitigates issues related to vanishing gradients and improves representation quality for longer inputs. The context and response encodings are compared using the same similarity scoring mechanism as in the RNN-based variant.

4.4 Evaluation Metrics

Response classification performance is evaluated using recall-based metrics. Given a dialogue context and a set of candidate responses, the model must identify the correct next utterance. Recall at k measures the proportion of test cases in which the correct response appears among the top k ranked candidates.

These metrics provide a straightforward and interpretable measure of model effectiveness in ranking tasks. They are particularly well suited for large-scale evaluation, where human judgments may be impractical to obtain for every example.

4.5 Experimental Results

Experimental results show that neural dual encoder models substantially outperform the TF-IDF baseline across all recall metrics. Among the neural approaches, the LSTM Dual Encoder achieves the strongest performance, reflecting its ability to model longer contextual dependencies more effectively.

Performance improves as the amount of dialogue context increases, indicating that the models benefit from access to longer conversation histories. However, gains diminish beyond a certain context length, suggesting limitations in the models' capacity to fully exploit very long sequences.

	Retrieval Metrics			
Method	1 in 2 R@1	1 in 10 R@1	1 in 10 R@2	1 in 10 R@5
TF-IDF	74.9%	48.8%	58.7%	76.3%
Dual Encoder w/RNN units	77.7%	37.9%	56.1%	83.6%
Dual Encoder w/LSTM units	**86.9%**	**55.2%**	**72.1%**	**92.4%**

Figure 8 Performance results of the three algorithms using recall metrics for binary (1-in-2) and 1-in-10 next utterance classification (%).

	Generative Metrics		
Method	Embedding Average	Greedy Matching	Vector Extrema
TF-IDF	0.536	0.370	0.342
Dual Encoder w/ LSTM units	**0.650**	**0.413**	**0.376**

Figure 9 Results for TF-IDF and the DE model with LSTM units, evaluated using embedding average, greedy matching, and vector extrema scores to assess the topic consistency of generated responses.

Context
"any apache hax around ? i just deleted all of __path__ - which package provides it ?", "reconfiguring apache do n't solve it ?"

Ranked Responses	Flag
1. "does n't seem to, no"	1
2. "you can log in but not transfer files ?"	0

Figure 10 An example displaying LSTM-ranked responses, with each utterance presented after pre-processing.

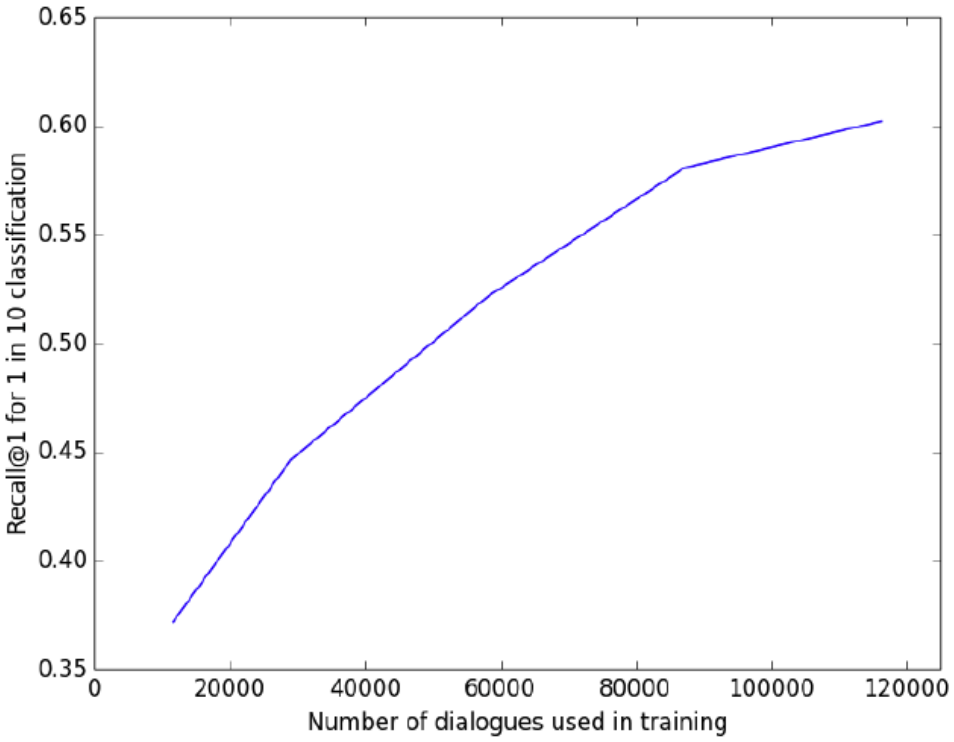

Figure 11 An LSTM with 200 hidden units showing Recall@1 for 1-in-10 classification as dataset size increases to 120k dialogues. Results are based on an earlier version of the Ubuntu Dialogue Corpus, so Recall@1 values are higher.

4.6 Qualitative Error Analysis

To better understand model behavior, a qualitative analysis of classification errors is conducted. Many errors arise in cases where multiple candidate responses are plausible given the dialogue context. In such situations, the model may rank a reasonable but incorrect response higher than the ground truth.

Other errors stem from limited world knowledge or insufficient understanding of technical details present in the dialogue. Ambiguous contexts and responses that depend on external information further complicate the task. These observations highlight the challenges inherent in response selection and suggest that improvements may require richer representations or access to additional contextual signals.

5. Generative Response Architectures

This section describes models designed to generate dialogue responses directly, rather than selecting them from a fixed set of candidates. These approaches model the

conditional probability of a response given a dialogue context and produce utterances token by token.

5.1 Generative Recurrent Neural Language Model

The generative recurrent neural language model defines a conditional probability distribution over a response sequence given the dialogue context. Let the dialogue context be represented as a sequence of tokens $c = (c_1, \dots, c_{|c|})$, and let the response be a sequence $r = (r_1, \dots, r_{|r|})$. The model factorizes the conditional probability of the response as

$$P(r \mid c) = \prod_{t=1}^{|r|} P(r_t \mid r_1, \dots, r_{t-1}, c)$$

The context tokens are first encoded into a fixed-dimensional vector using a recurrent neural network. This context representation initializes the hidden state of a second recurrent network responsible for generating the response. At each time step, the model updates its hidden state based on the previous token and produces a probability distribution over the vocabulary using a softmax layer.

Model parameters are learned by maximizing the conditional log-likelihood of the training responses given their corresponding contexts. This objective encourages the model to assign high probability to the observed responses in the dataset.

Context:	Speaker A: i can't seem to get audio working as a non-root user . has anyone ever had this problem ? _eou_ Speaker B: alsamixer to the rescue _eou_ Speaker A: alsamixer shows everything turned on , and looks exactly the same for my normal user as it does for root _eou_ Speaker B: no MM's? _eou_
Binary Probability	**Candidate Responses**
0.62	*correct _eou_*
0.92	true but how will he find my new ip so easily if i get it changed ? All i do is programming c and check my mail usually _eou_
0.68	yes strange . then omit the dash altogether , try giving set default sink :/ _eou_
0.93	**thanks _eou_ where is the db app - i cannot locate it (sorry to be such a noob !) _eou_**
0.03	I'm switching the location to my on board SSD drive that 's embedded to the laptops board . I just haven't been using the storage so I figure I could try and utilize the space while the ram being 8 gig 's itself I see no problem with the switch . Do you understand what I'm doing . I'm only asking here so I don't go screwing up and save myself hours of headaches _eou_
0.03	you'll love it _eou_ I am joking , but you will probably enjoy learning about it _eou_ well it 's a step up from opening your hard drive up and using a magnet _eou_
0.44	yeah , 512 is plenty _eou_
0.20	i use it on a number of machines with no problems . just this one . _eou_ modprobe pulls up a variety of mouse drivers _eou_
0.62	so the issue is **unknown** **unknown** . gz ' is different from the same file on the system " but i don't have any idea why/what that means , sorry . best of luck . _eou_
0.89	http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-680m comes with optimus technology . so i think it has an onboard intel card _eou_

Figure 12 An example where the model chooses an incorrect response to a question. The correct answer is shown in italics, while the model's selected answer is in bold.

Context:	Speaker A: hi _eou_ I have a problem with fn keys for brightness with my laptop and nvidia propertiary driver _eou_ Speaker B: what make and model laptop ? ¿ _eou_ Speaker A: sony vaio vgn fz31z _eou_ and im using nvidia propertiary driver version current (recommended one) _eou_ Speaker B: try the boot option : acpi_backlight=vendor _eou_ Speaker A: i have added acpi backlight for vendor i have updated grub but the keys are not working _eou_ my grub cmd line linxu default : quiet splash acpi_backlight=vendor _eou_ Speaker B: try the boot option : acpi_osi=LINUX _eou_ Speaker A: ok i must remove the acpi_backlihgt/ _eou_ Speaker B: I'd also report a bug _eou_ could try Quantal liveCD to see if the newer kernel plays nicer _eou_ Speaker A: I think that is a nvidia problem with the propertiary _eou_ Speaker B: possibly , or it could be acpi based _eou_ Speaker A: ok thank you i must remove the previous about the vendor ok ? _eou_ Speaker B: could try both and then just one _eou_
Binary Probability	**Candidate Responses**
0.38	*thanks for the help . Trying now . Is there any other same bug report for vaio/ _eou_*
0.20	it 's actually ubuntu support , since i'm using ubuntu , isn't it ? _eou_
0.49	yes _eou_ the usb disk will just be seen as a hard disk , install to it _eou_
0.50	if you do unattended-upgrades -d , that might tell you a few things ? _eou_
0.58	does this have ' open terminal here ' and ' 2pane mode ' options ? _eou_ found terminal option , just looking for 2pane _eou_
0.71	**it 's cool _eou_**
0.02	it 's like hotel internet _eou_ http://www.fdlinux.com/networksetuphowto.html _eou_
0.01	I'll check what it means in google . Thank you . _eou_
0.36	i never liked it ... for thin versions , i use fluxbox or some other window manager _eou_
0.49	so do I just paste that code in to the beginning of the script ... ? _eou_ sorry experienced linux user , very very novice coder ;-P _eou_

Figure 13 An example where the model fails to consider participant roles in the dialogue. The correct answer is in italics, while the model's selected answer is in bold.

Difficulty Rating (1-5)	**Number of Errors**	**% of Errors**
Impossible (5)	19	19%
Very difficult (4)	21	21%
Difficult (3)	22	22%
Moderate (2)	25	25%
Easy (1)	13	13%
Model Response Rating (1-3)		
Very reasonable (3)	14	14%
Somewhat reasonable (2)	37	37%
Unreasonable (1)	49	49%
Error Category		
Tone and style	8	9%
Knowledge	18	20%
Semantic similarity	45	49%
Word copying	11	12%
High-level inference	16	18%
Turn-taking structure	20	22%
Answering questions	6	7%

Figure 14 A qualitative analysis of errors in the DE model. Parent categories such as semantic similarity and turn-taking structure include their subcategories. Errors for impossible questions are not classified, and categories may overlap, so totals may not equal 100%.

5.2 Hierarchical Recurrent Encoder Decoder

The hierarchical recurrent encoder decoder extends the basic generative model by explicitly modeling dialogue structure at multiple levels. Instead of encoding the entire dialogue context as a single flat sequence, the context is represented as a sequence of utterances.

Each utterance in the context is encoded using an utterance-level recurrent encoder, producing a vector representation. These utterance vectors are then processed sequentially by a context-level recurrent network, which maintains a representation of the dialogue history across turns.

Given the context representation produced by the context-level network, a decoder recurrent network generates the response sequence. The conditional probability of the response is factorized as

$$P(r \mid c) = \prod_{t=1}^{|r|} P(r_t \mid r_1, \ldots, r_{t-1}, h_c)$$

where h_c denotes the hidden state of the context-level recurrent network summarizing the dialogue history. Training proceeds by maximizing the conditional log-likelihood of responses, analogous to the non-hierarchical model.

This hierarchical formulation allows the model to capture dependencies both within individual utterances and across multiple dialogue turns.

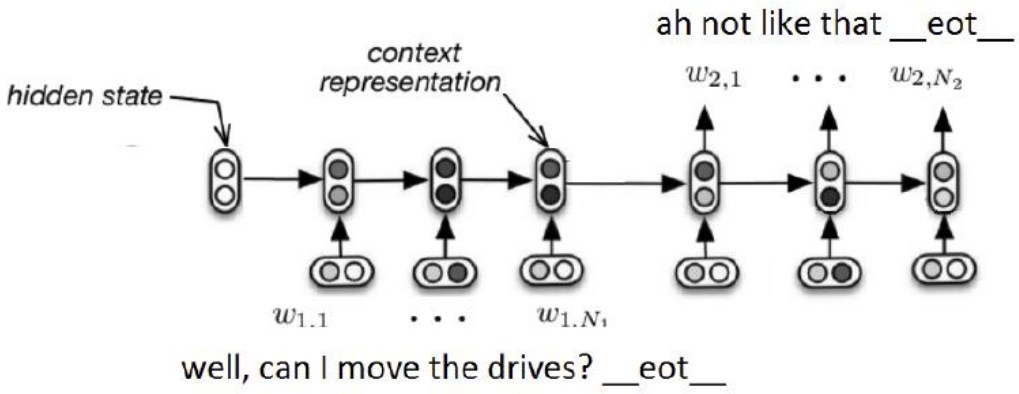

Figure 15 A diagram of the RNN architecture for dialogue modelling, where context utterances are concatenated before being input into the RNN.

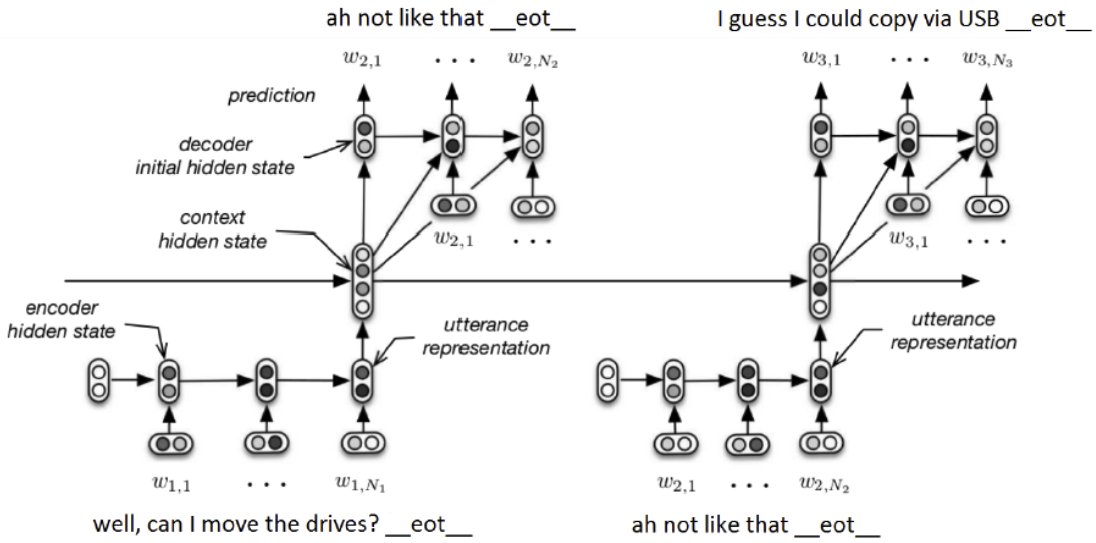

Figure 16 A diagram of the HRED model, where each context utterance is encoded by an utterance-level encoder and then passed to a context-level encoder.

	Generative Metrics		
	Embedding Average	Greedy Matching	Vector Extrema
LSTM-LM	0.561	0.425	0.380
HRED	0.617	**0.452**	**0.408**
TF-IDF	0.536	0.370	0.342
Dual Encoder w/ LSTM units	**0.650**	0.413	0.376

Figure 17 Results for both generative and retrieval models, evaluated using embedding average, greedy matching, and vector extrema scores to estimate the topic consistency of generated responses.

	Retrieval Metrics			
	1 in 2 R@1	1 in 10 R@1	1 in 10 R@2	1 in 10 R@5
LSTM-LM	58.9%	19.6%	33.1%	61.4%
HRED	**61.8%**	**21.5%**	**35.8%**	**64.5%**
TF-IDF	74.9%	48.8%	58.7%	76.3%
Dual Encoder w/RNN units	77.7%	37.9%	56.1%	83.6%
Dual Encoder w/LSTM units	86.9%	55.2%	72.1%	92.4%
MEMN2N (Dodge et al., 2015)	—	63.72%	—	—
RNN-CNN (Baudiš and Šedivỳ, 2016)	91.1%	67.2%	80.9%	95.6%
Ensemble (Kadlec et al., 2015)	**91.5%**	**68.3%**	**81.8%**	**95.7%**
r-LSTM (Xu et al., 2016)	88.9%	64.9%	78.5%	93.2%

Figure 18 Results for generative and retrieval models using recall metrics for binary (1-in-2) and 1-in-10 next utterance classification (%), including state-of-the-art results from recent studies.

5.3 Experimental Results

The generative models are evaluated using embedding-based similarity metrics that compare generated responses with ground-truth responses. These metrics assess semantic similarity rather than exact word overlap, which is important given the inherent variability of dialogue.

Results indicate that hierarchical generative models outperform flat recurrent language models across all evaluated metrics. The hierarchical encoder decoder generates longer responses with higher semantic similarity to the reference responses, suggesting improved use of dialogue context.

However, both models still struggle with maintaining precise technical correctness and avoiding generic replies in difficult cases. These limitations highlight the challenges of open-ended response generation in complex dialogue domains.

5.4 Examples of Generated Responses

Qualitative inspection of generated responses reveals clear differences between the models. The flat recurrent language model frequently produces short, generic replies that are broadly applicable but lack specific informational content.

In contrast, the hierarchical encoder decoder generates responses that are more detailed and contextually grounded. These responses often reference earlier parts of the dialogue and demonstrate better continuity across turns. At the same time, the increased specificity sometimes leads to minor inconsistencies or errors, reflecting a trade-off between diversity and safety in generative dialogue models.

Overall, the examples illustrate that explicitly modeling dialogue structure improves the richness and relevance of generated responses, even though further improvements are needed to achieve robust conversational performance.

Context	Response
1) i set up my hd such that i have to type a passphrase to access it at boot. how can i remove that passwrd, and just boot up normal. i did this at install, it works fine, just tired of having reboots where i need to be at terminal to type passwd in. help? → backup your data, and re-install without encryption " might " be the easiest method	**HRED:** how would i do that? **LSTM:** how do i do that?
2) hi, everyone. I download the googleearth. deb, and want to install it with : sudo dpkg -i something. deb, It tells me : google-earth-stable depends on lsb-core (>= 3.2) ; however : Package lsb-core is not installed. So i just install the lsb-core, But it comes error : <url> what should i do? → the error suggests running " sudo apt-get -f install " → i tried that, it comes the same error	**HRED:** try " sudo apt-get -f install " **LSTM:** <unk>
3) I've a 2.0 TB hard drive. It no longer boots from the hard drive, but I can see it from Gpart and it has a " boot " flag. How do I recover it? → the boot flag doesn't mean anything in Linux why do you not have a backup? you can mount the partition in liveCD and access the data easily (assuming the partition is healthy) → I have one hard drive for the moment (well, I don't count the 250GB drive) besides, I don't have two SATA cables I will make a backup as soon as I get another nice sized hard drive and a SATA cable And how do I find out if the partition is healthy? → boot to ubuntu livecd and try and mount the partition	**HRED:** How do I do that? **LSTM:** on the live cd?
4) bottom line is usually if you got 64bit why not use it? (unless there is something that doesn't work on 64bit, which is amitedly not much) → well, now Clip2net is not working... just seems Ubuntu runs more seamlessly on 32-bit and after reading this article, Im wondering if 64 is a bit hyped? <url>	**HRED:** I have no experience with 64bit, so I don't know what you mean. **LSTM:** <unk>, <unk>, <unk>, <unk>, <unk>
5) How would I enable vim syntax highlighting on a default install of ubuntu 12.04? → have you insalled vim instead of the default vim-tiny? → I installed vim-runtime but still no highlighting	**HRED:** yes **LSTM:** yes
6) Its comming from my computer 15 people using same router and they have no issues, also it happened recently, before it was ok → yes but your router may have optional settings for dns which if not set to the same dns will override your network interface... it 's an option to consider	**HRED:** how do I do that? **LSTM:** that is what I was looking for, thanks
7) it looks as if the network 's DHCP server is dead : " < warn > (eth0) : DHCPv4 request timed out " → but it isn't. windows works	**HRED:** that 's what I thought, but I don't know how to fix it **LSTM:**?
8) I've been having trouble with a new 3TB harddrive and a fresh Ubuntu install. I install (as GPT) and the install looks ok. But then get GRUB " error : out of disk " after reboot. I have an October 2011 PC with updated BIOS and I _do_ see the 3TB. google searches haven't been to helpful so far. has anyone seen GRUB trouble when installing on 3TB harddrives? → have you tried installing grub again and updating it? sudo grub-install sudo update-grub? → I guess I could try that. So I would have to install GRUB from an older Ubuntu install?	**HRED:** yes, you can install grub to the MBR of your hard drive **LSTM:** grub is installed on the mbr of the hard drive

Figure 19 Examples from the Ubuntu model, where → denotes a turn change and <unk> represents an unknown token.

6. Discussion

This study examined end-to-end neural dialogue models trained on large-scale, unstructured conversational data. By evaluating both response selection and response generation approaches on the Ubuntu Dialogue Corpus, the analysis highlights the strengths and limitations of current architectures. While neural models are capable of learning meaningful conversational patterns directly from data, several open challenges remain, particularly in handling complex conversational structure, evaluation, and generalization beyond observed interactions.

6.1 Conversation Disentanglement

One of the central challenges posed by the Ubuntu Dialogue Corpus is conversation disentanglement. The original chat logs are drawn from multi-user channels in which several conversations may occur simultaneously. Although preprocessing heuristics are applied to extract dyadic dialogues, residual entanglement can still occur. This introduces noise into both training and evaluation, as utterances may be incorrectly attributed to a dialogue context.

Improving disentanglement methods would likely benefit downstream dialogue modeling. More sophisticated approaches could incorporate learned representations of speaker interaction patterns or temporal dynamics to better separate overlapping conversations. Addressing this issue is an important step toward improving the quality and reliability of large-scale dialogue datasets derived from real-world chat environments.

6.2 Non-Task Oriented Model Evaluation

Evaluating dialogue systems that are not strictly task-oriented remains an open problem. In domains such as technical support, there may be multiple valid responses to a given context, each differing in style, level of detail, or problem-solving strategy. This inherent variability makes it difficult to define a single correct output.

Human evaluation provides valuable insight into response quality, but it is expensive and difficult to scale. Moreover, judgments can vary depending on the evaluator's background knowledge, particularly in technical domains. These factors complicate direct comparisons between models and motivate the need for complementary evaluation approaches.

6.3 Automatic Evaluation of Dialogue Systems

Automatic metrics commonly used in natural language generation, such as word overlap measures, have limited effectiveness for dialogue evaluation. Because conversational responses can be phrased in many semantically equivalent ways, low lexical overlap does not necessarily indicate poor quality. Embedding-based similarity metrics offer some improvement by capturing semantic relatedness, but they still correlate weakly with human judgments.

The results presented in this work reinforce the need for better automatic evaluation methods tailored specifically to dialogue. Future metrics should account for contextual appropriateness, informativeness, and coherence across multiple turns, rather than relying solely on surface-level similarity to reference responses.

6.4 Future Research Directions for End-to-End Systems

The findings of this study suggest several directions for future research. First, incorporating richer memory mechanisms or external knowledge sources may help models handle complex technical content and long-term dependencies more effectively. Second, improved modeling of uncertainty and response diversity could reduce the prevalence of generic replies while maintaining coherence.

Additional progress may also come from improved dataset construction and annotation strategies, including better disentanglement and the inclusion of auxiliary supervision signals. Finally, advances in evaluation methodology will be essential for accurately measuring progress and guiding model development. Together, these directions point toward more robust and scalable end-to-end dialogue systems capable of operating in realistic conversational environments.

References

[1] Bayer, J. and Osendorfer, C. Learning stochastic recurrent networks. In *Advances in Variational Inference Workshop*, NIPS, 2014.

[2] Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. Modeling temporal dependencies in high-dimensional sequences with application to polyphonic music generation and transcription. In *Proceedings of the International Conference on Machine Learning*, 2012.

[3] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[4] Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. One billion word benchmark for measuring progress in statistical language modeling. In *Proceedings of INTERSPEECH*, 2014.

[5] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder decoder architectures for statistical machine translation. In *Proceedings of EMNLP*, 2014.

[6] Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. A recurrent latent variable model for sequential data. In *Proceedings of NeurIPS*, 2015.

[7] Fabius, O. and van Amersfoort, J. R. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

[8] Galley, M., Brockett, C., Sordoni, A., Ji, Y., Auli, M., Quirk, C., Mitchell, M., Gao, J., and Dolan, B. DeltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. In *Proceedings of ACL*, 2015.

[9] Goodfellow, I., Courville, A., and Bengio, Y. *Deep Learning*. MIT Press, 2015.

[10] Graves, A. Sequence transduction with recurrent neural networks. In *Proceedings of the ICML Representation Learning Workshop*, 2012.

[11] Gregor, K., Danihelka, I., Graves, A., and Wierstra, D. DRAW: A recurrent neural network for image generation. In *Proceedings of ICLR*, 2015.

[12] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8), 1997.

[13] Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, 2015.

[14] Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *Proceedings of ICLR*, 2014.

[15] Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL*, 2016.

[16] Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., and Pineau, J. How not to evaluate your dialogue system. *arXiv preprint arXiv:1603.08023*, 2016.

[17] Lowe, R., Pow, N., Serban, I., and Pineau, J. The Ubuntu Dialogue Corpus. In *Proceedings of SIGDIAL*, 2015.

[18] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. Recurrent neural network based language model. In *Proceedings of INTERSPEECH*, 2010.

[19] Mitchell, J. and Lapata, M. Vector-based models of semantic composition. In *Proceedings of ACL*, 2008.

[20] Pietquin, O. and Hastie, H. A survey on metrics for the evaluation of user simulations. *The Knowledge Engineering Review*, 28(1), 2013.

[21] Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of ICML*, 2014.

[22] Ritter, A., Cherry, C., and Dolan, W. B. Data-driven response generation in social media. In *Proceedings of EMNLP*, 2011.

[23] Rus, V. and Lintean, M. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the ACL Workshop on Building Educational Applications*, 2012.

[24] Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., and Pineau, J. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of AAAI*, 2016.

[25] Shaikh, S., Strzalkowski, T., Taylor, S., and Webb, N. VCA: An experiment with a multiparty virtual chat agent. In *Proceedings of the ACL Workshop on Companionable Dialogue Systems*, 2010.

[26] Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Simonsen, J. G., and Nie, J.-Y. A hierarchical recurrent encoder decoder for generative context-aware query suggestion. In *Proceedings of CIKM*, 2015.

[27] Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of NAACL-HLT*, 2015.

[28] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Proceedings of NeurIPS*, 2014.

[29] Young, S., Gasic, M., Thomson, B., and Williams, J. D. POMDP-based statistical spoken dialogue systems: A review. *IEEE Proceedings*, 101(5), 2013.