

Do Encoder Representations of Generative Dialogue Models Encode Sufficient Information about the Task ?

Prasanna Parthasarathi^{1,2}, Joelle Pineau^{1,2,4}, Sarath Chandar^{2,3,4}

¹ School of Computer Science, McGill University

² Quebec Artificial Intelligence Institute (Mila), Canada

³ École Polytechnique de Montréal

⁴ Canada CIFAR AI Chair

Abstract

Predicting the next utterance in dialogue is contingent on encoding of users' input text to generate appropriate and relevant response in data-driven approaches. Although the semantic and syntactic quality of the language generated is evaluated, more often than not, the encoded representation of input is not evaluated. As the representation of the encoder is essential for predicting the appropriate response, evaluation of encoder representation is a challenging yet important problem. In this work, we showcase evaluating the text generated through human or automatic metrics is not sufficient to appropriately evaluate soundness of the language understanding of dialogue models and, to that end, propose a set of probe tasks to evaluate encoder representation of different language encoders commonly used in dialogue models. From experiments, we observe that some of the probe tasks are easier and some are harder for even sophisticated model architectures to learn. And, through experiments we observe that RNN based architectures have lower performance on automatic metrics on text generation than transformer model but perform better than the transformer model on the probe tasks indicating that RNNs might preserve task information better than the Transformers.

1 Introduction

The task of dialogue modeling requires learning through interaction, often, from humans. The model is expected to understand the input text for it to interact, and the interaction can be meaningful only when the language understanding gets better. Approaches for solving dialogue task include information retrieval based approaches like selecting a response from a set of canned responses (Lowe et al., 2015a) or keeping track of very specific information which are *a priori* marked as informative slot-value pairs (Guo et al., 2018; Asri

et al., 2017) or generating the next response with token-by-token (Vinyals and Le, 2015; Lowe et al., 2015a; Serban et al., 2015; Li et al., 2016, 2017; Parthasarathi and Pineau, 2018). The evaluation of the different approaches have mostly relied on the output of the model – the slot predicted, response selected or generated.

The issues in evaluation – automatic evaluation metrics uncorrelated with human judgement – showcased by Liu et al. (2016) is still an open problem. Attempts to mimic human scores for better evaluation metric (Lowe et al., 2017) and other metrics that aim to correlate with the human judgement (Sinha et al., 2020; Tao et al., 2018) evaluate the quality of the text generated but do not evaluate the language understanding component of a model. The language understanding component of an agent more often than not goes unnoticed with only token-level evaluation metrics on the generated text.

To that end, we propose evaluating the encoder representation of dialogue models through probe tasks¹ constructed from the commonly used dialogue data sets – MultiWoZ (Budzianowski et al., 2018) and PersonaChat (Zhang et al., 2018). Concretely, we use the representation learnt by the encoders while training on dialogue generation tasks to solve a set of dialogue related classification tasks as a proxy to probe the information encoded in the encoder representation. We study the performance of language encoders in 17 different probe tasks with varying degree of difficulties – binary classification, multi-label classification and multi-label prediction. For example, predicting whether the current dialogue has single or multiple tasks, identifying the number of tasks, identifying the tasks, presence of a specific information provided by the user among many others. The probe tasks allow

¹<https://github.com/ppartha03/Dialogue-Probe-Tasks-Public>

a way to quantify the understanding of a model and help identify biases, if any, in the task of dialogue prediction. We observed the performance of the models in the probe tasks to little fluctuate with different seed values thus allowing to analyse the encoder representation with minimal variance. Further, the experiments on probe tasks help in understanding deeper differences in between recurrent neural network (RNN) and Transformer encoders that were previously not evident from the token-level evaluation methods.

Our contributions in the paper are:

- Showcasing the significantly high variance in human evaluation of dialogues.
- Proposing a list of probe tasks – 2 semantic, 13 information specific and 3 downstream as an alternate evaluation of dialogue systems.
- Finding that the representation learnt by recurrent neural network based models is better at solving the probe tasks than the ones based on transformer model.

2 Related Work

Evaluating dialogue models has been an important topic of study. While many of the metrics have focussed on evaluating the generated text through n-gram overlap based heuristics – BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Lavie and Agarwal, 2007) – there have also been learned metrics like ADEM (Lowe et al., 2017), MAudE (Sinha et al., 2020), RUBER (Tao et al., 2018) among other metrics (Celikyilmaz et al., 2020). Though language generation has been an important component of study, there are not many studies that benchmark soundness of encoding information by dialogue systems.

Probe tasks in language generation (Conneau et al., 2018; Belinkov and Glass, 2019; Elazar et al., 2020) has been used to understand the information encoded in continuous embedding of sentences. Such probe tasks are set up as classification tasks that are solved with model learnt representation. As it is easier to control the biases in probe tasks than in the downstream tasks, research in language generation has analysed models on probe tasks like using encoder representation to identify words in input (**WordCont**) to measuring encoder sensitivity to shifts in bigrams (Conneau et al., 2018; Belinkov and Glass, 2019).

Analysis using probe tasks has been done also in reinforcement learning (RL). Anand et al. (2019) learn state representation for an RL agent in an unsupervised setting and introduce a set of probe tasks to evaluate the representation learnt by agents. This includes using an annotated data set with markers for position of the agent, current score, items in inventory, target’s location among others. The authors train a shallow linear classifier to identify specific entities in the embedded input that serves as a metric for the representational soundness of the learning algorithm.

Applications of computer vision like caption generation for images (Vinyals et al., 2015) or videos (Donahue et al., 2015) use attention based models to parse over the hidden states of a convolutional neural network (ConvNet) (LeCun et al., 1998). The attention over the ConvNet features are visualized to observe the words corresponding to different parts of the image. Visualizing the attention has been one of the qualitative probe task for text generation conditioned on images (Xu et al., 2015).

3 Dialogue probe Tasks

Like other tasks, dialogue task requires a learning agent to have sufficient understanding of the context to generate a response; at times the models have been shown to not have basic understanding leading to incorrect response prediction. Although dialogue models are evaluated on grammar, semantics, and relevance of the generated text, seldom has that been extended to evaluate the language encoding capacity of these models. The tasks proposed and discussed in this paper are shown in Table 1.

3.1 Basic Probe Tasks

The basic probe tasks evaluate if the encoder representation can be used to predict the existence of a mid-frequency token in the context (*WordCont*) (Belinkov and Glass, 2019), or test if the encoding of the context provides information of how long the dialogue has been going on (*UtteranceLoc*) (Sinha et al., 2020). For *UtteranceLoc* task, the conversation is split into 5 different temporal blocks and a classifier trained on the encoded context embedding is used to predict the appropriate label.

3.2 Information Specific Probe Tasks

We construct 12 information specific probe tasks to evaluate if specific information is retained in the encoder representation of input text. The informa-

Task	Task Name	Description	#Classes	Multi-Label Prediction
Semantic	UtteranceLoc *	How long has the conversation been happening ?	5	No
	WordCont ⁺	Which mid-frequency word is encoded in the context ?	1000	No
Information Specific	IsMultiTopic	Does the conversation have more than one topic ?	2	No
	NumAllTopics	How many topics does this conversation have ?	6	No
	RepeatInfo	Which information provided by the user is repeated ?	11	Yes
	NumRepeatInfo	What many number of recent information are repeats ?	7	No
	AllTopics	What are all the topics discussed so far ?	8	No
	RecentSlots	What is the <i>recent</i> information given by the user ?	37	Yes
	NumRecentInfo	How <i>many</i> information did the user provide <i>recently</i> ?	10	No
	RecentValues	What are the details of the <i>recent</i> information ?	1060	Yes
	AllSlots	What <i>all</i> information are given by the user so far?	37	Yes
	AllValues	What are the details of <i>all</i> the information provided ?	1060	Yes
	RecentTopic	What is the current topic of the dialogue ?	8	No
Downstream task	NumAllInfo	How <i>many</i> information did the user provide so far ?	20	No
	PersonalInfo ⁺	What keywords in USER persona does the model identify?	3754	Yes
	ActionSelect	Which downstream task (database query) follows the current conversation ?	32	No
	EntitySlots	What information is required to construct the query ?	29	Yes
	EntityValues	What values should be passed to the query ?	1309	Yes

Table 1: The difficulty levels of different tasks is measured with the average performance of an untrained encoder. There is a natural grading in the selection of tasks that expects better language understanding to solve. ⁺ indicate the task is present both in MultiWoZ and PersonaChat datasets. * indicate the task is only in PersonaChat. If no indicator is present, the task is evaluated only in MultiWoZ dataset.

tion specific tasks have different levels of difficulty. For example, *IsMultiTopic* is a binary classification task, *NumAllTopics* is a multi-label classification task while *AllTopics* is a multi-label prediction.

3.3 Downstream probe Tasks

Further we evaluate the language understanding of dialogue models on their performance on relevant downstream tasks. Towards evaluating the model’s understanding of the user utterance, the downstream probe tasks verify if the encoder representation allows to predict the user dialogue act. The dialogue state tracking measures the performance of a model on such tasks (Henderson et al., 2014) but seldom is it evaluated on generative dialogue models. Neelakantan et al. (2019) use entity, values and action information to train on the dialogue generation task but the performance of a generative dialogue model without explicitly training on the downstream tasks are not compared. Towards that, we propose **ActionSelect**, **EntitySlots**, **EntityValues** probe tasks. The details of the task are shown in Table 1.

4 Experiments

4.1 Data sets

With the probe tasks we study different dialogue encoder architectures trained on next utterance generation on MultiWoZ 2.0 (Budzianowski et al., 2018)

and PersonaChat (Zhang et al., 2018) data sets. The features of the data sets are shown in Table 2. To

Data set	Train	Validation	Vocabulary
<i>PersonaChat</i>	~ 10900	1500	16k
<i>MultiWoZ</i>	~ 8400	1000	13k

Table 2: Distribution of the dialogues in the data sets.

comprehensively compare several model selection criteria, we experimented with selecting models based on BLEU (Papineni et al., 2002), ROUGE-F1 (Lin, 2004), METEOR (Lavie and Agarwal, 2007) and Vector-Based (Average BERT embedding) metrics. We present the results from BLEU as a selection criteria in the paper. Further in the Appendix we compare the evolution of the performance of different models in the probe tasks over the entire training.

The classification tasks for probing the encoder representation are constructed for every generated response that requires information from the dialogue history thus far. We split the probe tasks in Train/Test/Valid corresponding to the splits the tasks are constructed from. First, we train the dialogue models on end-to-end dialogue generation and use the encoder representation to train and test on the probe tasks. To that, we store the encoder parameters after every epoch during dialogue generation training and compute the results of probe tasks after every epoch.

4.2 Models

We train 5 commonly used encoder architectures on the task of next utterance generation in the two data sets.

LSTM ENCODER-DECODER The architecture (Vinyals and Le, 2015) has an LSTM cell to encode the input context only in the forward direction. For a sequence of words in the input context $(w_1^i, w_2^i, \dots, w_T^i)$ LSTM encoder generates $\{h_t\}_1^T$. The decoder LSTM’s hidden state is initialized with h_t^T and the decoder outputs one token at each step of decoding. For the experiments, we used two layer LSTM cell where the first layer applies recurrent operation on the input to the model and the layer above recurs on the outputs of the layer below. The encoder final hidden state (from the 2nd layer) is passed as an input to the decoder. We train the model with cross entropy loss as shown in Equation 1.

$$\sum_{t=1}^T -y_t \log(p(\hat{y}_t)) - (1 - y_t) \log(1 - p(\hat{y}_t)) \quad (1)$$

where y_t is the t^{th} ground truth token distribution in the output sequence, \hat{y}_t is model generated token and p is the model learned distribution over the tokens. We train the model with Adam (Kingma and Ba, 2014) optimizer with teacher forcing (Williams and Zipser, 1989).

LSTM ENCODER-ATTENTION DECODER

The architecture is similar to the LSTM Encoder-Decoder with an exception of an attention module to the decoder. The attention module (Bahdanau et al., 2014) linearly combines the encoder hidden states $h_{t_1}^T$ as an input to the decoder LSTM at every step of decoding, unlike only having the last encoder hidden state.

HIERARCHICAL RECURRENT ENCODER DECODER

The model has encoding done by two encoder modules acting at different levels (Sordoni et al., 2015); *sentence encoder* to encode the sentences that feeds in as input to the *context encoder*. Both the encoders are LSTMs. The decoder is an attention decoder.

BI-LSTM ENCODER-ATTENTION DECODER

The encoder is a concatenation of two LSTMs that can read the input from forward and backward direction (Schuster and Paliwal, 1997). The hidden state is computed as the summation of the hidden

states of the two encoders. The decoding is done with an attention decoder.

TRANSFORMER ARCHITECTURE This state-of-the-art architecture (Vaswani et al., 2017; Rush, 2018) is a transductive model that has multiple layers of attention to predict the output. We used the architecture in an encoder-decoder style by splitting half the layers for encoding and the remainder for decoding. We perform the probe tasks on the encoder hidden state computed as an average over word token attention.

The size of the models used in the experiments are detailed in Table 7 in Appendix. For the probe tasks, we select the untrained model, model with the best BLEU score on validation, and model from the last training epoch. We use packages pytorch (Paszke et al., 2017) and scikit-learn (Pedregosa et al., 2011) for our experiments.

4.3 Motivation for Dialogue Probe Tasks

The texts generated by the models are largely dependent on the choice of seed values and a slight variation could result in a model generating a very different response. Although the automatic metrics have greater agreement on the score across seed values, we see that human participants do not agree on the consistency of the generated response. We pose and evaluate an alternate hypothesis where we expect the participants to identify two responses to be similar when selected from different runs of the same model with different seed values that have similar BLEU scores.

Model	PersonaChat	MultiWoZ
BiLSTM + Attn	4.4 ± 0.06	15.5 ± 0.05
Seq2Seq	4.5 ± 0.06	15.8 ± 0.17
Seq2Seq + Attn	4.4 ± 0.15	15.7 ± 0.11
HRED	3.9 ± 0.01	12.2 ± 4.00
Transformer	7.9 ± 0.17	29.4 ± 0.61

Table 3: BLEU scores of the models from runs with different seeds on PersonaChat and MultiWoZ data set. (Higher the better. We measure BLEU-2 (case insensitive)).

For the study, we sample 2000 context-response pairs in MultiWoZ dataset from the model with lower variance in BLEU score (Table 3) – BiLSTM Attention – from its two different runs. We ask the participants to select the response that is more *relevant* to the given context, similar to Li et al. (2015). The annotators can select either of the responses or a Tie². For every context-response

²The human evaluation proposal was evaluated and ap-

pair, we collect 3 feedback from different participants (Distribution corresponding to the 3 different human responses are shown with legend HumanExp1, HumanExp2 and HumanExp3 in Figure 1).

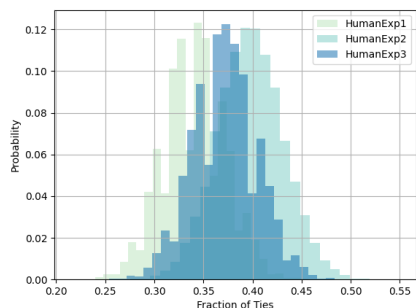


Figure 1: The mean of the distribution of tie in three different experiments was centered around 35%, showing that the subjective scores on responses by humans are not sufficient to evaluate a model.

Usually human evaluation is done on 100-500 responses. To understand the variance in this set up and the lack of information at the token generation level, we sample 50000 sets of 200 human responses from the collected 2000 responses and compute the fraction of times there was a tie. We observed that distribution over the fraction of times the human participants selected a *Tie* was centered around 35% (Figure 1) with all of the probability mass within 50%. This shows that (a) text generated by the same model produce significantly different responses with different seed values (b) attributing the choice of seed value to the performance of a model creates confusion in the evaluation because the two seeds had similar BLEU scores. The results show that evaluating only based on the text generated by a model is not suggestive of the information encoding capacity of the encoder representation. Also, the dependence of the model generated text on seed value raises a valid concern; whether a model parameter initialized with a specific seed value mimic the token generation of a model that actually encodes sufficient information in the context. The lack of clarity leads to inconclusiveness of studies with human evaluation to show whether the dialogue models have sufficient information encoded to solve the task effectively.

proved by an IRB.

4.4 Probe Tasks

We train the models with the two dialogue data sets on next utterance generation. To understand the evolution on the probe task, we compare with 3 different parameter configurations of every model – *Untrained*, *Last epoch*, and *BestBLEU*. We use Logistic Regression classifier³ implementation from scikit-learn (Pedregosa et al., 2011) with default parameters except the `max_iter` set to 250 for all the probe tasks. The evaluation metric is *F1*-score with micro averaging in multi-class prediction tasks.

PROBE TASKS ON PERSONACHAT The models are evaluated on three probe tasks (Table 4) – two basic and one information specific. *Utterance-Loc* and *WordCont* measures if the encoded context suggests semantic awareness of the model while *PersonalInfo* measures the amount of knowledge the model has about its persona from encoding of conversation history. In other words, it evaluates the extent to which persona can be identified from the context encoding with a linear classifier. A better performance in these tasks indicate that the context encoding preserves information on persona and the temporal order of the dialogue.

The *PersonalInfo* task is not very specific to identifying personal information but acts as an indicator to the information embedded in dialogues that goes unnoticed in the encoding. It was surprising to see that no model scored a reasonable *F1*. Although Transformer model scored higher on BLEU, (Table 3) the performance of transformer on *PersonalInfo* task was decreasing throughout the training epochs (Table 4).

The tasks *UtteranceLoc* and *WordCont* evaluate if encoder representations are indicative of how far in the conversation is the model in and identify mid-frequency words in the target response respectively. Bi-LSTM model performed the best in *Utterance-Loc* while the Transformer model was not in the top 3.

We observe that the inductive biases of the RNN-based models enable random projections that are informative even without training. This correlates with independent observations on the results in (Tallec et al., 2019) that argues random projections of temporal information hold non-negligible information. Similar observations are also made from the untrained Transformer model’s performance on the

³Also, we trained a nonlinear model –multi-layer perceptron for probe tasks and the results are similar. The discussion in the paper is agnostic to the choice of the classifier.

PersonaChat data set			
Model	UtteranceLoc	WordCont	PersonalInfo
Bi-LSTM Seq2Seq + Attention			
Untrained	37.0 ± 0.1	43.5 ± 0.0	0.0 ± 0.0
LastEpoch	56.5 ± 0.0	39.9 ± 0.0	0.0 ± 0.0
BestBLEU	57.2 ± 0.1	39.7 ± 0.1	0.0 ± 0.0
HRED - LSTM			
Untrained	1.2 ± 0.0	51.7 ± 0.0	0.0 ± 0.0
LastEpoch	12.8 ± 4.9	49.4 ± 0.3	0.0 ± 0.0
BestBLEU	10.8 ± 3.5	51.0 ± 0.1	0.0 ± 0.0
LSTM Seq2Seq + Attention			
Untrained	39.9 ± 0.0	47.2 ± 0.1	0.0 ± 0.0
LastEpoch	52.0 ± 0.0	40.0 ± 0.0	0.0 ± 0.0
BestBLEU	54.1 ± 0.16	43.8 ± 0.2	0.0 ± 0.0
LSTM Seq2Seq			
Untrained	40.2 ± 0.0	46.9 ± 0.0	0.0 ± 0.0
LastEpoch	50.9 ± 0.1	40.0 ± 0.0	0.0 ± 0.0
BestBLEU	52.2 ± 0.1	40.2 ± 0.0	0.0 ± 0.0
Transformer Architecture			
Untrained	53.0 ± 0.0	35.9 ± 0.0	2.4 ± 0.0
LastEpoch	42.7 ± 0.1	46.9 ± 0.1	0.0 ± 0.0
BestBLEU	40.7 ± 0.1	46.2 ± 0.0	0.0 ± 0.0

Table 4: Performance of different models on the probe tasks on PersonaChat data set. The performance is measured as $F-I$ score (Higher the better).

probe tasks.

The RNN encoders project the context to a smaller manifold with its recurrent multiplication that regularizes its representation to observe structures, whereas Transformer network’s attention operations project the context on to a larger manifold that prevents loss in encoding⁴ making the representation useful for the end task (Figure 2). This explains the RNN based encoders performing well on UtteranceLoc while Transformer model performing well on WordCont. The difference between the two classes of models is much more evident on the probe tasks in MultiWoZ data set.

PROBE TASKS ON MULTIWOZ In majority of information specific tasks and in the downstream tasks (Table 5), we observed that RNN based models perform significantly better than the Transformer model. Interestingly, we observed a pattern in Transformer in the two data sets, that the model’s performance on the probe tasks decreased from the beginning of training till the end on all of the tasks, while for the rest of the models there was learning involved.

The downsampled encoder representation of the encoded contexts with PCA to 2 components (Figure 2) shows that the range of the two axes are different for RNN-based and Transformer models. The context encoding of transformers lie in a much larger manifold. The attention layers help in spreading the data in a large manifold thereby the model

⁴Ramsauer et al. (2020) showed recently that the transformer model is a large look-up table. Our empirical results support the authors’ view.

can retain almost all of the generation task related information it was trained on. This can be observed in higher BLEU score the model achieves in language generation. But, the reverse of generalizing from a small data is hard to come by because the model does not have sufficient direct information to cluster except the surface level signal of predicting the right tokens. This helps the Transformer model to perform well on the token prediction task in language modelling, while abstracting information and generalizing appears to be a difficult task as is observed from its performance on probe tasks.

The RNN-based models have inductive biases to squish the input through \tanh or sigmoid operations. From the visualizations and from other results, we hypothesize that this aids the model in learning a regularized representation in a low-data set up. But, this can potentially be unhelpful when the input is a large set of samples and has rich structure as that requires a model to aggressively spread out. Transformer architecture can thrive in such a set up and that can be validated by the performance of large Transformer models like GPT (Radford et al., 2018), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), BERT (Devlin et al., 2018, 2019), RoBERTa (Liu et al., 2019) etc., whereas the results in the probe tasks show that RNN-based models are adept at learning unsupervised structures for better understanding of the input. Also we note that the performance in probe tasks can be a pseudo metric to measure the capacity of the model in generalizing to unobserved structures in inputs in a low data scenario.

5 Discussion

Systematic evaluation of language understanding through probe tasks is important to analyze the correlation between input and output in complex language understanding tasks. We observed that most of the data collected for dialogue generation tasks (Lowe et al., 2015b; Ritter et al., 2011) do not provide tasks to sanity check language understanding through probing encoder representations. Absence of probe tasks lead to draw imperfect correlations like the one between token-level accuracy and model’s encoding of dialogue information from the context. At this point one may wonder, why not train the model with all the probe tasks as auxiliary tasks for an improved performance? Although it is a possibility, such a set up does not evaluate a model’s ability to generalize to understanding in

(a)

MultiWoZ data set								
Model	UtteranceLoc	RecentTopic	RecentSlots	RecentValues	RepeatInfo	NumRepeat	NumRecent	AllSlots
LSTM Seq2Seq + Attention								
Untrained	46.5 ± 0.5	35.3 ± 0.0	39.1 ± 0.0	30.8 ± 0.0	64.2 ± 0.0	69.0 ± 0.0	41.4 ± 0.0	30.2 ± 0.0
LastEpoch	56.5 ± 0.1	87.1 ± 0.0	65.6 ± 0.0	42.2 ± 0.0	64.9 ± 0.0	70.0 ± 0.0	61.7 ± 0.1	51.5 ± 0.0
BestBLEU	58.0 ± 0.1	89.0 ± 0.0	66.5 ± 0.0	41.1 ± 0.0	64.5 ± 0.0	67.0 ± 0.0	63.4 ± 0.0	52.0 ± 0.1
HRED - LSTM								
Untrained	45.3 ± 1.6	32.9 ± 0.0	41.2 ± 0.0	31.7 ± 0.0	71.0 ± 0.0	74.9 ± 0.0	40.7 ± 0.0	19.8 ± 0.0
LastEpoch	38.0 ± 10.9	54.2 ± 22.6	36.3 ± 10.1	21.3 ± 3.4	69.4 ± 0.1	74.0 ± 0.0	39.5 ± 11.7	32.8 ± 8.4
BestBLEU	38.7 ± 11.3	50.1 ± 20.5	34.3 ± 9.3	20.4 ± 3.1	71.0 ± 0.1	74.5 ± 0.1	39.3 ± 11.6	30.3 ± 7.9
LSTM Seq2Seq								
Untrained	46.6 ± 0.3	35.9 ± 0.0	39.7 ± 0.0	32.0 ± 0.0	64.8 ± 0.0	69.2 ± 0.0	43.0 ± 0.0	29.5 ± 0.0
LastEpoch	55.0 ± 0.1	87.6 ± 0.0	66.0 ± 0.0	41.9 ± 0.0	66.1 ± 0.0	69.8 ± 0.0	61.0 ± 0.0	51.6 ± 0.0
BestBLEU	56.3 ± 0.0	88.6 ± 0.0	66.9 ± 0.0	41.6 ± 0.0	65.9 ± 0.0	70.2 ± 0.0	62.6 ± 0.0	52.6 ± 0.0
Bi-LSTM Seq2Seq + Attention								
Untrained	44.3 ± 0.0	50.7 ± 0.1	35.3 ± 0.0	27.3 ± 0.0	64.6 ± 0.0	70.6 ± 0.0	39.9 ± 0.0	36.7 ± 0.0
LastEpoch	57.2 ± 0.0	86.7 ± 0.0	63.3 ± 0.0	38.2 ± 0.0	66.6 ± 0.0	70.8 ± 0.0	60.2 ± 0.1	53.4 ± 0.0
BestBLEU	57.5 ± 0.1	89.0 ± 0.0	64.5 ± 0.0	39.6 ± 0.0	68.5 ± 0.0	72.2 ± 0.0	62.3 ± 0.0	56.0 ± 0.0
Transformer Architecture								
Untrained	51.2 ± 0.0	80.3 ± 0.0	45.6 ± 0.0	30.6 ± 0.0	70.4 ± 0.0	73.3 ± 0.0	47.5 ± 0.0	62.9 ± 0.0
LastEpoch	33.7 ± 0.5	32.1 ± 1.9	26.2 ± 1.9	22.1 ± 1.7	70.7 ± 0.0	74.6 ± 0.0	33.6 ± 3.3	21.3 ± 0.6
BestBLEU	32.0 ± 0.5	31.7 ± 5.3	29.6 ± 0.3	25.3 ± 0.2	72.2 ± 0.0	75.9 ± 0.0	37.8 ± 0.4	22.8 ± 1.44

(b)

MultiWoZ data set								
Model	AllValues	NumAllInfo	AllTopics	NumAllTopics	IsMultiTask	EntitySlots	EntityValues	ActionSelect
LSTM Seq2Seq + Attention								
Untrained	12.6 ± 0.0	7.0 ± 0.0	45.1 ± 0.0	70.3 ± 0.0	80.1 ± 0.0	28.0 ± 0.0	19.6 ± 0.0	28.7 ± 0.0
LastEpoch	19.3 ± 0.0	29.3 ± 0.0	73.4 ± 0.0	76.3 ± 0.0	81.5 ± 0.0	43.5 ± 0.0	28.4 ± 0.0	56.2 ± 0.0
BestBLEU	18.7 ± 0.0	29.2 ± 0.0	74.3 ± 0.0	76.9 ± 0.1	82.1 ± 0.0	42.6 ± 0.0	29.1 ± 0.0	56.9 ± 0.0
HRED - LSTM								
Untrained	5.3 ± 0.0	0.0 ± 0.0	37.5 ± 0.0	77.6 ± 0.0	84.2 ± 0.0	24.9 ± 0.1	19.0 ± 0.0	27.3 ± 0.01
LastEpoch	8.7 ± 0.7	19.1 ± 2.8	48.7 ± 18.0	69.2 ± 3.7	73.5 ± 4.7	27.1 ± 5.6	20.2 ± 3.1	38.8 ± 11.3
BestBLEU	8.4 ± 0.8	18.0 ± 2.6	46.6 ± 16.9	68.6 ± 3.5	73.2 ± 4.6	24.8 ± 4.9	20.1 ± 3.0	34.4 ± 9.3
LSTM Seq2Seq								
Untrained	13.3 ± 0.0	6.3 ± 0.0	43.0 ± 0.0	73.3 ± 0.0	80.4 ± 0.1	27.3 ± 0.0	20.3 ± 0.0	29.0 ± 0.0
LastEpoch	19.5 ± 0.0	28.8 ± 0.0	72.8 ± 0.0	75.7 ± 0.0	81.2 ± 0.0	44.0 ± 0.0	30.7 ± 0.0	56.7 ± 0.0
BestBLEU	18.8 ± 0.00	29.7 ± 0.02	74.3 ± 0.03	77.1 ± 0.0	81.9 ± 0.0	44.1 ± 0.0	28.9 ± 0.03	57.2 ± 0.0
Bi-LSTM Seq2Seq + Attention								
Untrained	14.9 ± 0.0	10.9 ± 0.1	56.8 ± 0.0	71.4 ± 0.0	79.5 ± 0.0	24.2 ± 0.0	19.0 ± 0.0	26.1 ± 0.0
LastEpoch	20.0 ± 0.0	28.5 ± 0.0	74.8 ± 0.0	78.4 ± 0.0	84.0 ± 0.0	42.1 ± 0.0	29.6 ± 0.0	55.4 ± 0.0
BestBLEU	20.0 ± 0.0	29.6 ± 0.0	77.4 ± 0.0	79.1 ± 0.0	84.2 ± 0.0	41.6 ± 0.0	28.2 ± 0.0	56.5 ± 0.0
Transformer Architecture								
Untrained	39.6 ± 0.0	27.3 ± 0.0	81.2 ± 0.0	77.6 ± 0.0	82.8 ± 0.0	30.3 ± 0.0	19.7 ± 0.1	38.5 ± 0.2
LastEpoch	5.1 ± 0.1	11.5 ± 0.5	47.7 ± 1.3	71.9 ± 0.0	82.0 ± 0.0	13.5 ± 0.4	13.4 ± 0.0	6.8 ± 0.1
BestBLEU	5.6 ± 0.1	7.3 ± 0.2	50.4 ± 1.1	73.5 ± 0.0	81.7 ± 0.0	23.3 ± 0.1	12.2 ± 0.3	7.8 ± 0.2

Table 5: F1 scores of generative dialogue models on probe tasks in MultiWoZ dialogue data set (higher the better). SEQ2SEQ models perform significantly better than Transformer model on the probe tasks, despite the models falling behind in BLEU score. The Transformer model’s performance decreased from initial to last epoch in majority of the tasks while SEQ2SEQ models have a learning curve.

unseen dialogues. One could potentially train a model with a fraction of the probe tasks as auxiliary tasks and evaluate on the rest, we leave that for future work.

It is also interesting to draw parallels to *Unit Testing* in software engineering (Koomen and Pol, 1999), where the smallest software components of a system are tested for their design and logical accuracy. The difference between a deterministic application software and a stochastic decision making ML module is that the behavior of the ML system is data-driven while for a software system it is driven by logic. Despite the difference, the unit testing and probe tasks could share a common ground towards ensuring the better representation of the encoded contexts.

Model	Easy	Medium	Hard
<i>LSTM-Attn</i>	77.6±6.2	65.7±7.6	44.4±23.7
<i>HRED</i>	72.1±2.7	39.3±5.1	25.4±13.6
<i>Seq2Seq</i>	77.2±5.3	65.7±7.6	44.9±23.5
<i>BiLSTM</i>	78.5±6.2	65.6±8.7	44.2±23.3
<i>Transformer</i>	77.2±4.9	43.3±14.7	24.4±16.4

Table 6: Aggregate F1 scores of the models on performance in probe tasks on MultiWoZ data set.

DIALOGUE MODELS As an alternate to token-level evaluation, comparison of different model architectures can be meaningfully made with an aggregate metric on the probe tasks in three groups of difficulty – *easy* ((Ave. SEQ2SEQ) Untrained F1 > .50), *medium* (0.25 < Untrained F1 ≤ 0.50), and *hard* (Untrained F1 < 0.25). Such an analysis, as shown in Table 6, allows better inspection of

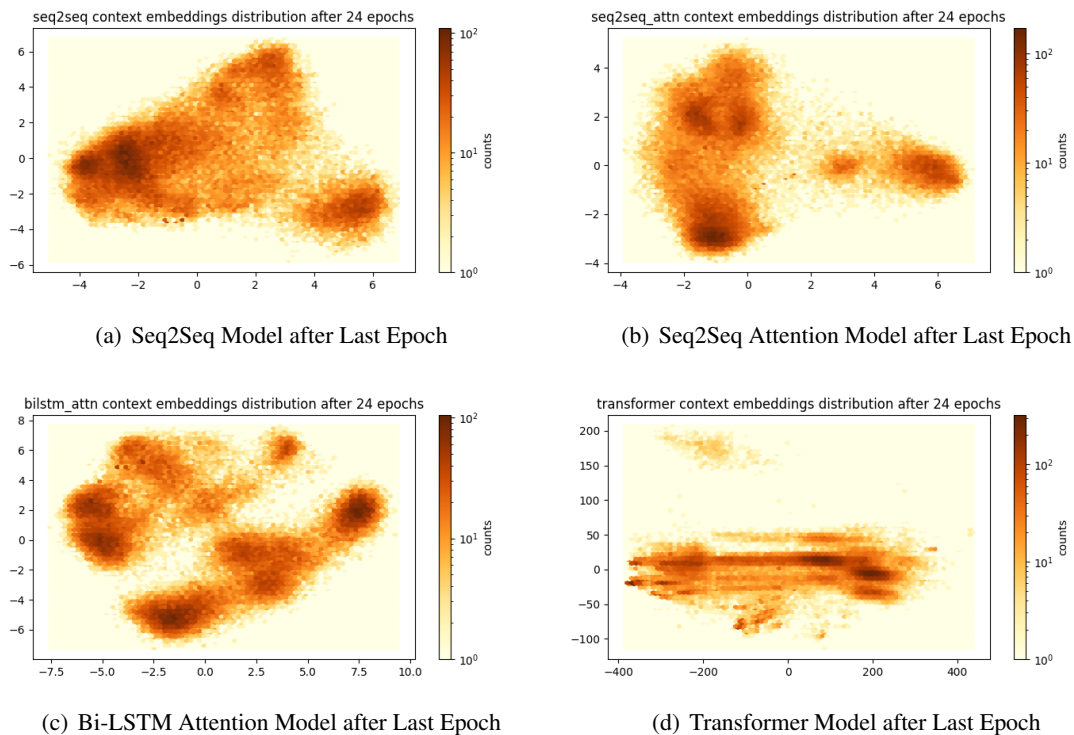


Figure 2: Downsampled encoder hidden states on MultiWoZ data set with PCA show that Transformer model has high capacity to encode a large data set unlike the SEQ2SEQ models.

the model’s language understanding and a fairer comparison between the models. We can see from Table 6 that the models have difficulty in learning to solve hard probe tasks from the encoder representations. The results can be used to build novel inductive biases for neural architectures that address one or a group of aspects in the language understanding of dialogue prediction models.

DIALOGUE DATA SETS The challenges in dialogue modeling has been evolving majorly because of the complex data sets. But, data sets on chit-chat dialogues often have little to no auxiliary tasks to evaluate the dialogue management abilities of a model. This limits the practitioners to validate the models only on the text generation task which, in this paper, is shown to have little to no correlation with the model’s ability to understanding the encoded summary of natural language context.

6 Conclusion

We propose a set of probe tasks to evaluate the encoder representation of end-to-end generative dialogue models. We observed that mimicking surface level token prediction do not reveal much about a model’s ability to understand a natural language context. The results on probe tasks showed

that RNN-based models perform better than transformer model in encoding information in the context. We also found some probe tasks that all of the models find difficult to solve; this invites novel architectures that can handle the language understanding aspects in dialogue generation. Although language generation is required for a dialogue model, the performance in token/response prediction alone cannot be a proxy for the model’s ability to understand a conversation. Hence, systematically identifying issues in language understanding through probe tasks can help in building better models and collecting challenging data sets.

Acknowledgements

We would like to acknowledge Compute Canada and Calcul Quebec for providing computing resources used in this work. The authors would also like to thank members of Chandar Research Lab, Mila for helping with the code reviews and reviewing the manuscripts. Sarath Chandar and Joelle Pineau are supported by Canada CIFAR AI Chair, and Sarath Chandar is also supported by an NSERC Discovery Grant.

References

- Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. 2019. Unsupervised state representation learning in atari. In *NeurIPS*.
- Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. *arXiv*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *arXiv*.
- Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *TACL*.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv*.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of EMNLP*.
- Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv*.
- Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \mathbb{R}^d vector: Probing sentence embeddings for linguistic properties. In *Proceedings of ACL*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.
- Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. When bert forgets how to pos: Amnesic probing of linguistic properties and mlm predictions. *arXiv*.
- Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-action: Conversational question answering over a large-scale knowledge base. In *Advances in Neural Information Processing Systems*.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *SIGDIAL*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv*.
- Tim Koomen and Martin Pol. 1999. *Test process improvement: a practical step-by-step guide to structured testing*. Addison-Wesley Longman Publishing Co., Inc.
- Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv*.
- Jiwei Li, Will Monroe, Alan Ritter, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. *Proceedings of EMNLP*.
- Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *arXiv*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.
- Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of EMNLP*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv*.
- Ryan Lowe, Michael Noseworthy, Iulian V Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv*.
- Ryan Lowe, Nissan Pow, Iulian Serban, Laurent Charlin, and Joelle Pineau. 2015a. Incorporating unstructured textual knowledge sources into neural dialogue systems. In *In NeurIPS Workshop on Machine Learning for Spoken Language Understanding*.

- Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015b. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *arXiv*.
- Arvind Neelakantan, Semih Yavuz, Sharan Narang, Vishaal Prasad, Ben Goodrich, Daniel Duckworth, Chinnadhurai Sankar, and Xifeng Yan. 2019. Neural assistant: Joint action prediction, response generation, and latent knowledge reasoning. *arXiv*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics.
- Prasanna Parthasarathi and Joelle Pineau. 2018. Extending neural generative conversational model using external knowledge sources. In *Proceedings of EMNLP*.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. *NIPS-W*.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.
- Hubert Ramsauer, Bernhard Schöfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, et al. 2020. Hopfield networks is all you need. *arXiv*.
- Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*.
- Alexander M Rush. 2018. The annotated transformer. In *Proceedings of workshop for NLP open source software (NLP-OSS)*.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. In *IEEE Transactions on Signal Processing*.
- Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. Hierarchical neural network generative models for movie dialogues. *arXiv*.
- Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L Hamilton, and Joelle Pineau. 2020. Learning an unreferenced metric for online dialogue evaluation. In *Proceedings of ACL*.
- Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*.
- Corentin Tallec, L'aronard Blier, and Diviyani Kalainathan. 2019. Reproducing "world models": Is training the recurrent network really needed? <https://ctaltec.github.io/world-models/>.
- Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*.
- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv*.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*.
- Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv*.

Model	Parameters
<i>LSTM Encoder-Decoder</i>	11M
<i>LSTM Encoder-Decoder + Attention</i>	11M
<i>HRED</i>	12M
<i>Bi-LSTM Encoder-Decoder</i>	12M
<i>Transformer</i>	41M

Table 7: Size of parameters of the models used in all the experiments on the two data sets. M for Million.

Appendix

A Model Parameters

- For SEQ2SEQ models, we used a 256 unit hidden size LSTM with 2 layers and a 128 unit input embedding dimension. The learning rate we used for all the models is 4E-3.
- For Transformer, we used a 512 unit hidden size, 512 unit input embedding dimension, 2 attention header and 4 layers.
- We used Adam as the optimizer to optimize on the cross-entropy loss.
- We averaged the results over 3 different seeds.
- We used a truncated history of last 100 tokens as context to keep the training uniform across the models.

MultiWoZ Dataset								
Model	UtteranceLoc	RecentTopic	RecentSlots	RecentValues	RepeatInfo	NumRepeatInfo	NumRecentInfo	AllSlots
LSTM Seq2Seq + Attention								
BERT	37.12 ± 2.59	42.74 ± 16.78	43.53 ± 4.54	30.93 ± 0.63	70.82 ± 0.01	74.71 ± 0.01	44.76 ± 1.89	23.94 ± 6.31
F1	58.08 ± 0.09	89.31 ± 0.08	66.72 ± 0.02	39.55 ± 0.05	71.25 ± 0.01	75.10 ± 0.00	62.14 ± 0.02	52.57 ± 0.10
BLEU	57.55 ± 0.05	89.91 ± 0.07	67.39 ± 0.02	40.49 ± 0.04	70.92 ± 0.00	74.73 ± 0.00	62.48 ± 0.02	53.08 ± 0.11
METEOR	58.23 ± 0.08	89.26 ± 0.08	66.83 ± 0.02	39.72 ± 0.04	71.29 ± 0.00	75.01 ± 0.00	62.23 ± 0.01	52.58 ± 0.10
HRED - LSTM								
BERT	18.78 ± 10.58	23.78 ± 16.97	16.41 ± 8.07	10.44 ± 3.27	71.78 ± 0.02	75.51 ± 0.01	19.27 ± 11.14	13.31 ± 5.31
F1	37.18 ± 10.38	49.59 ± 19.55	33.95 ± 8.98	20.81 ± 3.26	71.33 ± 0.01	74.99 ± 0.01	38.49 ± 11.14	28.49 ± 6.63
BLEU	37.15 ± 10.35	50.98 ± 20.94	34.84 ± 9.69	20.63 ± 3.21	71.68 ± 0.00	75.06 ± 0.00	38.59 ± 11.18	30.23 ± 7.84
METEOR	41.04 ± 5.85	50.78 ± 20.86	44.50 ± 2.49	28.96 ± 0.18	71.72 ± 0.00	75.28 ± 0.00	50.71 ± 1.44	30.21 ± 7.84
LSTM Seq2Seq								
BERT	54.16 ± 0.94	63.24 ± 16.20	55.13 ± 4.78	34.62 ± 0.76	72.00 ± 0.00	75.90 ± 0.00	54.06 ± 2.46	37.48 ± 5.21
F1	57.56 ± 0.06	89.44 ± 0.04	68.00 ± 0.00	40.98 ± 0.03	71.22 ± 0.01	75.32 ± 0.01	62.78 ± 0.01	53.07 ± 0.04
BLEU	57.37 ± 0.06	89.45 ± 0.03	68.08 ± 0.01	39.78 ± 0.07	71.28 ± 0.01	75.36 ± 0.01	62.33 ± 0.05	53.40 ± 0.05
METEOR	57.84 ± 0.04	89.03 ± 0.01	67.74 ± 0.01	40.37 ± 0.10	71.10 ± 0.00	74.75 ± 0.00	61.85 ± 0.00	53.04 ± 0.02
Bi-LSTM Seq2Seq + Attention								
BERT	57.98 ± 0.03	78.79 ± 3.19	57.24 ± 1.71	35.59 ± 0.34	71.35 ± 0.00	75.18 ± 0.01	57.57 ± 0.18	48.37 ± 1.11
F1	57.99 ± 0.05	89.63 ± 0.03	64.85 ± 0.00	39.16 ± 0.00	71.76 ± 0.01	75.30 ± 0.01	60.85 ± 0.07	54.68 ± 0.04
BLEU	59.04 ± 0.10	89.85 ± 0.03	65.03 ± 0.00	39.06 ± 0.00	71.98 ± 0.01	75.63 ± 0.00	60.36 ± 0.05	54.96 ± 0.05
METEOR	58.45 ± 0.07	89.28 ± 0.02	64.21 ± 0.00	39.19 ± 0.00	71.54 ± 0.00	75.35 ± 0.01	60.49 ± 0.05	54.65 ± 0.04
Transformer Architecture								
BERT	39.11 ± 0.09	58.38 ± 0.14	29.97 ± 0.00	24.50 ± 0.01	72.39 ± 0.01	76.02 ± 0.00	38.80 ± 0.01	43.37 ± 0.17
F1	39.89 ± 0.21	67.44 ± 0.44	33.37 ± 0.14	24.96 ± 0.02	72.75 ± 0.01	76.26 ± 0.00	40.43 ± 0.05	51.19 ± 0.51
BLEU	39.46 ± 0.00	57.05 ± 1.50	30.10 ± 0.27	23.72 ± 0.03	72.70 ± 0.00	75.97 ± 0.00	39.11 ± 0.08	40.43 ± 1.21
METEOR	38.50 ± 0.25	56.26 ± 1.87	30.98 ± 0.11	24.94 ± 0.02	72.26 ± 0.01	75.79 ± 0.00	39.47 ± 0.04	38.70 ± 1.59

Table 8: Comparison of models selected different selection metrics on probe tasks in MultiWoZ dialogue data set. The performance is measured with $F1$ on the probetasks.

MultiWoZ Dataset								
Metric	AllValues	NumAllInfo	AllTopics	NumAllTopics	IsMultiTask	EntitySlots	EntityValues	ActionSelect
LSTM Seq2Seq + Attention								
BERT	6.16 ± 0.34	8.52 ± 2.18	49.07 ± 5.13	77.98 ± 0.00	84.97 ± 0.01	27.49 ± 1.30	22.22 ± 0.47	30.25 ± 6.74
F1	12.54 ± 0.01	26.54 ± 0.02	75.22 ± 0.03	79.56 ± 0.02	84.70 ± 0.01	41.74 ± 0.02	31.20 ± 0.03	60.00 ± 0.00
BestBLEU	12.81 ± 0.01	25.73 ± 0.02	75.33 ± 0.02	79.39 ± 0.02	85.30 ± 0.00	41.29 ± 0.03	31.57 ± 0.03	60.14 ± 0.01
METEOR	12.53 ± 0.01	26.62 ± 0.02	75.21 ± 0.03	79.52 ± 0.02	84.67 ± 0.01	41.70 ± 0.02	31.48 ± 0.02	60.06 ± 0.00
HRED - LSTM								
BERT	3.20 ± 0.31	7.49 ± 1.68	21.92 ± 14.41	58.94 ± 3.05	62.30 ± 4.46	10.85 ± 3.53	9.06 ± 2.46	17.04 ± 8.72
F1	6.40 ± 0.32	16.07 ± 1.97	45.79 ± 16.07	69.01 ± 3.62	73.72 ± 4.79	23.39 ± 4.22	19.53 ± 2.87	35.39 ± 9.73
BLEU	6.90 ± 0.39	14.96 ± 1.77	46.63 ± 16.93	68.66 ± 3.50	72.97 ± 4.50	24.33 ± 4.64	19.97 ± 3.01	35.66 ± 9.95
METEOR	6.82 ± 0.38	15.93 ± 1.93	54.09 ± 8.47	79.20 ± 0.02	85.55 ± 0.01	30.35 ± 1.29	25.88 ± 0.51	36.00 ± 9.70
LSTM Seq2Seq								
BERT	9.16 ± 0.27	18.10 ± 2.47	60.55 ± 4.58	77.91 ± 0.03	84.43 ± 0.02	34.68 ± 1.90	27.23 ± 0.79	45.12 ± 7.11
F1	12.92 ± 0.01	26.47 ± 0.04	74.63 ± 0.03	78.44 ± 0.00	84.05 ± 0.01	43.66 ± 0.01	31.83 ± 0.01	61.11 ± 0.01
BLEU	12.76 ± 0.01	26.94 ± 0.04	75.03 ± 0.03	78.16 ± 0.00	83.90 ± 0.00	43.92 ± 0.01	31.96 ± 0.01	61.13 ± 0.00
METEOR	12.97 ± 0.00	25.97 ± 0.03	74.37 ± 0.01	78.42 ± 0.00	84.03 ± 0.01	43.79 ± 0.04	31.63 ± 0.02	61.22 ± 0.02
Bi-LSTM Seq2Seq + Attention								
BERT	12.83 ± 0.10	23.74 ± 0.13	71.48 ± 1.07	78.54 ± 0.07	85.60 ± 0.00	35.96 ± 0.72	26.88 ± 0.07	50.57 ± 1.36
F1	14.92 ± 0.00	26.67 ± 0.07	78.01 ± 0.01	81.02 ± 0.06	86.17 ± 0.00	40.61 ± 0.00	29.38 ± 0.01	57.91 ± 0.01
BLEU	15.13 ± 0.01	25.87 ± 0.05	78.11 ± 0.02	80.43 ± 0.02	86.20 ± 0.00	40.82 ± 0.01	29.91 ± 0.02	57.76 ± 0.00
METEOR	14.81 ± 0.00	26.53 ± 0.07	78.04 ± 0.01	80.02 ± 0.01	86.25 ± 0.00	41.02 ± 0.00	30.11 ± 0.02	57.90 ± 0.01
Transformer Architecture								
BERT	11.81 ± 0.04	9.01 ± 0.06	65.01 ± 0.09	76.23 ± 0.02	84.38 ± 0.01	20.60 ± 0.00	18.87 ± 0.02	15.48 ± 0.14
F1	17.97 ± 0.64	11.26 ± 0.17	71.08 ± 0.24	77.82 ± 0.03	85.27 ± 0.01	22.47 ± 0.02	19.06 ± 0.03	20.24 ± 0.34
BestBLEU	10.43 ± 0.14	9.71 ± 0.00	64.42 ± 0.88	76.10 ± 0.07	84.20 ± 0.01	19.83 ± 0.00	18.34 ± 0.03	15.35 ± 0.54
METEOR	10.77 ± 0.37	7.92 ± 0.11	63.64 ± 0.80	76.58 ± 0.05	84.50 ± 0.01	20.17 ± 0.06	18.38 ± 0.01	15.03 ± 0.72

Table 9: Comparison of models selected different selection metrics on probe tasks in MultiWoZ dialogue data set. The performance is measured with $F1$ on the probetasks.