

# IEC: Towards Interest-Eliciting Neural Conversational Agents

Ziyu Yao<sup>\*\*</sup> Yizhe Zhang<sup>†</sup> Xiujun Li<sup>†§</sup> Jianfeng Gao<sup>†</sup>  
Michel Galley<sup>†</sup> Chris Brockett<sup>†</sup> Huan Sun<sup>\*</sup> Bill Dolan<sup>†</sup>

<sup>\*</sup>The Ohio State University <sup>§</sup>University of Washington

<sup>†</sup>Microsoft Research AI

{yao.470, sun.397}@osu.edu {yizhe.zhang, xiul, jfgao, mgalley, Chris.Brockett, billdol}@microsoft.com

## Abstract

Conversational agents need sufficient social skills to establish long-term connections with human users. **Identifying users' interests without explicitly asking about them, such as in salesperson-customer conversations, is among the most important social skills.** In this paper we formulate the problem as an *Interest-Eliciting Conversation (IEC)* task, where a conversational agent must elicit users' interests and shift to topics that can effectively drive the conversation forward. We present a first-of-its-kind research testbed with a neural user simulator and a set of competitive baseline agents. Some of these agents are equipped with a neural decision module to intelligently elicit users' interests so as to sustainably anticipate and respond to their needs. We empirically demonstrate the effectiveness of the IEC agents and lay the foundation for the development of future conversational agents that are more user-engaging.

## 1 Introduction

Building conversational agents that are able to engage in empathetic conversations with humans has been one of the long-running goals in natural language processing (NLP) communities (Gao et al. 2019; Zhou et al. 2018). One major challenge in building such conversational agents is how to improve long-term user engagement (Ram et al. 2018; Fang et al. 2018). Recent studies have shown that a user's interest in a topic is significantly, positively correlated with the resulting user engagement (Glas, Prepin, and Pelachaud 2015; Glas and Pelachaud 2015). **In many scenarios, however, users may not explicitly express their interests before establishing trust or emotional bond with their conversation partners.** For instance, in salesperson-customer interactions such as product promotion and business negotiation, a customer tends to evade communications and conceals her true attitudes when she does not have enough trust in the salesperson (Schurr and Ozanne 1985; Swan and Nolan 1985). Explicitly inquiring about user interests, e.g., by asking "what do you like", may therefore run counter to our purpose.

In this paper, we study the problem of *Interest-Eliciting Conversation (IEC)*, where a conversational agent seeks to

<sup>\*</sup>Manuscript. Work done while interning at MSR AI.

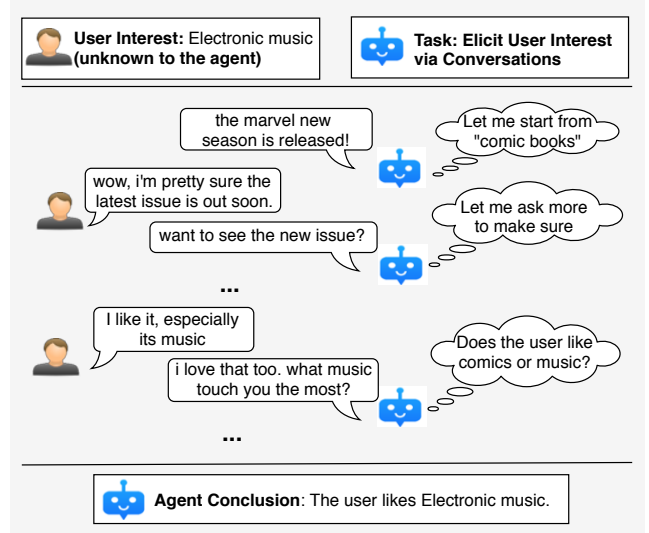


Figure 1: An illustration of the Interest-Eliciting Conversation (IEC) task.

elicit the user's interest and intelligently switch topics to drive the conversation. As illustrated in Figure 1, the user has an underlying topic of interest (e.g., electronic music) that is unknown to the bot. **The job of the bot is to infer the user's implicitly expressed interests without explicitly asking them: the bot needs to detect the user's implicit preference in each utterance and changes in preference over time (e.g., from marvel comics to its music), then steer the conversation topic to meet or trigger the user's interest (e.g., talking about music).** Success in proactive interest elicitation is likely not only to enhance users' experience with the bot in scenarios as sales recommendation (where the bot might occasionally suggest services based on the dialogue context), but more fundamentally to help the agent to establish long-term emotional connections with the user, which in turn allows the agent to provide more personalized services that better meet the user's needs.

To this end, we present the first attempt to systematically explore the IEC problem. *First*, we present a *simplified* and *controlled* simulated user-bot interaction setting as a first-of-

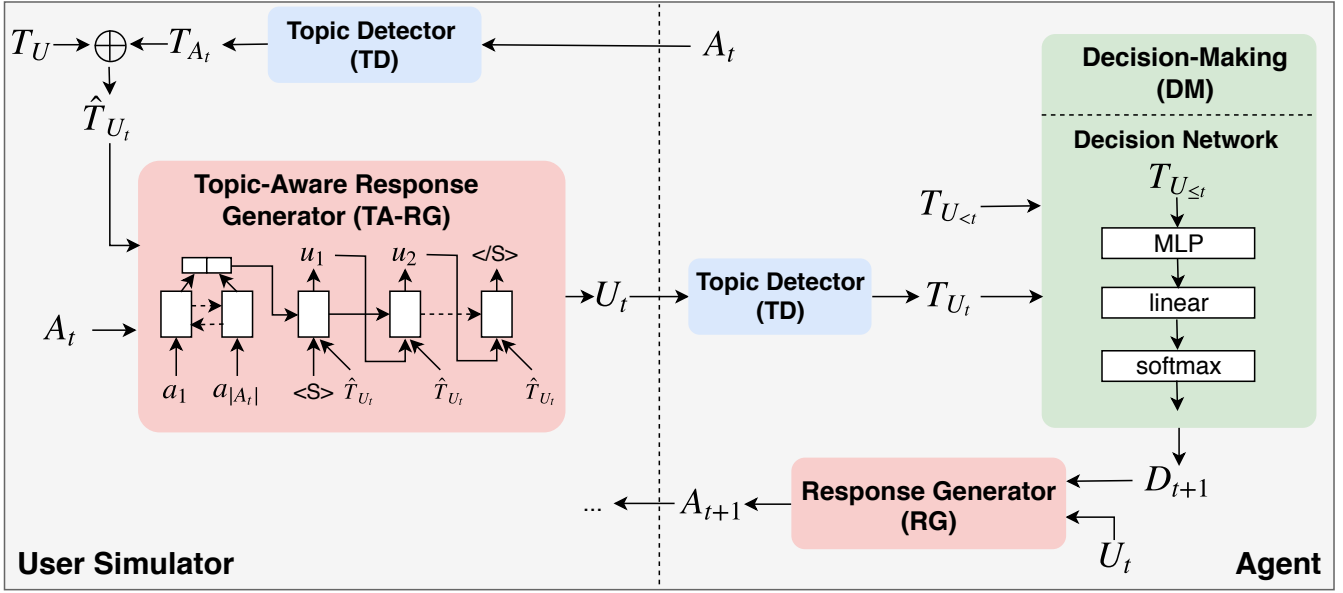


Figure 2: The Interest-Eliciting Conversation (IEC) framework. **User Simulator:** Given an agent utterance  $A_t$ , a *Topic Detector* first identifies its topic distribution  $T_{A_t}$ . The distribution is then combined with the true user interest  $T_U$  and fed to a *Topic-Aware Response Generator* to generate the user response  $U_t$ . **Agent:** The agent decides the next topic  $D_{t+1}$  by feeding the topics of the user’s previous utterances ( $T_{U_{\leq t}}$ ) to a *Decision-Making* module. A response generator further takes in  $D_{t+1}$  and the user utterance  $U_t$  to produce an agent response  $A_{t+1}$ .

its-kind testbed for the IEC problem (Figure 2). The testbed consists of a virtual environmental platform, a set of system evaluation criteria, and a neural user simulator, which generates simulated user utterances according to a pre-specified topic of interest and a set of rules that specify the simulator’s behaviors (e.g., **leaving the conversation when receiving dull agent utterances**). Although these specifications make the simulator’s behaviors unrealistic in some cases, our setting simplifies the real-world IEC task, thus providing a more achievable goal as a stepping stone for researchers to work on. *Second*, based on the simulated environment, we develop several baseline conversational agents. They differ in the use of neural or template-based response generators and different decision-making strategies of driving conversations. The most sophisticated of these IEC agents uses a neural decision network that proactively adapts the conversation topic to meet the user’s interest. We also show how to evaluate the effectiveness of these agents using the simulated IEC setting.

## 2 Related Work

Our work marries the goal-oriented dialog with the social chat conversation. For conversations, one end of the spectrum is **goal-oriented dialog systems** (Williams and Young 2007; Henderson, Thomson, and Williams 2014; Wen et al. 2017; Li et al. 2018), where agents are trained towards optimizing an objective. Such agents have been applied to bargain negotiation (Lewis et al. 2017; He et al. 2018), object guessing (De Vries et al. 2017; Das et al. 2017), movie promotion (Yu, Black, and Rudnicky 2017) and museum guide (Glas, Prepin, and Pelachaud 2015). For

instance, Lewis et al. (2017) investigated negotiation conversations among two agents to reach an agreement that can maximize their respective rewards. Yu, Black, and Rudnicky (2017) explored the integration of chit chat and task-oriented dialogs to promote movies. Our work is different from the above: (1) Our agent is designed to be *fully data-driven* and *end-to-end (E2E)*. (2) While previous work are usually restricted in specific domains such as movies (Yu, Black, and Rudnicky 2017) and museums (Glas, Prepin, and Pelachaud 2015), we allow *open-domain* conversations without making further assumptions.

At the other end of the spectrum, is **social chat-bots** (Vinyals and Le 2015; Sordoni et al. 2015; Li et al. 2016b; Zhang et al. 2018b). To encourage consistent responses, Li et al.; Mazare et al.; Zhang et al. (2016a; 2018; 2018a) further encoded personality in the response generator. Compared with existing work which usually train the bot at a *sentence level* (e.g., optimizing response diversity), we assign a *long-term goal* for our agent to drive the entire conversation, which potentially improves the user engagement. Moreover, this goal can serve as a new evaluation metric from a long-term perspective. Similar to our work, Misu and Kawahara; Yoshino and Kawahara (2007; 2015) constructed agents to keep track of user interests during conversations for information navigation. **Our task is different from theirs in that we focus on a salesperson-customer scenario where users do not explicitly show their interests before engaging into the conversation.** Tang et al.; Wu et al.; Qin et al. (2019; 2019; 2020) also explored goal-driven open-domain chat-bots, but their goals are specified keywords or entities in a

knowledge graph, which is different from ours.

Finally, due to the data scarcity issue, user simulator has been widely used in task-oriented dialogue systems (Li et al. 2016c; El Asri, He, and Suleman 2016; Kreyssig et al. 2018). **In comparison, our user model is more sophisticated in that it can generate topic-aware responses under multiple topics, and the generator is based on neural network trained on a large scale of conversations.**

### 3 User Model

To introduce the IEC framework, we start from the neural conversational user simulator in Figure 2. We consider a simplified scenario where the user is assumed to be interested in a single topic that has not been revealed to the agent. However, the framework can be easily extended to cases where users have multiple interests. As illustrated in Figure 2, the conversation is initiated by the agent. Given the current agent utterance  $A_t$ , the user simulator first detects the *agent topic*  $T_{A_t}$  via a *Topic Detector* (Section 3.2). A user response  $U_t$  is then generated by a *Topic-Aware Response Generator* (Section 3.3) based on  $A_t$ , its topic  $T_{A_t}$  and the user’s own interest (Section 3.1). This procedure is iterated for a maximum number of turns or until the user ends the conversation. Without loss of generality, we pre-define  $K = 10$  topics in our experimental setting.

#### 3.1 User Behavior Simulation

To yield tractable inference and to ease evaluation, we consider several heuristic specifications for the user simulator. We note such a user simulator may not necessarily reflect real user behaviors, nonetheless, it can facilitate the development and validation of an IEC agent in a controlled manner. Specifically, each simulated user is designed to maintain a user interest distribution  $T_U$  of size  $K$ . Since we assume a user has only one topic of interest,  $T_U$  is a one-hot vector.

In conversation, humans tend to be coherent with the topic in the conversation history and also to reveal their own interests. To emulate this behavior, we adopt a linear interpolation between the *contextual topic distribution*  $T_{A_t}$ , and the *user interest distribution*  $T_U$ , as the *target topic distribution* ( $\hat{T}_{U_t}$ ) of the user’s response, i.e.:

$$\hat{T}_{U_t} = \lambda T_U + (1 - \lambda) T_{A_t}, \lambda \in [0, 1], \quad (1)$$

where  $T_{A_t}$  is captured by the topic detector (Section 3.2) based on the current agent utterance  $A_t$ .

We also model the user losing interest when the agent has discussed a less interesting topic several times. We consider the most prominent topic (indexed by  $\arg\max_{i \in [1, K]} (T_{A_t})_i$ ) as the *major topic* of an agent utterance  $A_t$ . If a topic (different from the user’s interest) has been the major topic for  $S$  times (in our experiments  $S = 3$ ), we decay its value in  $T_{A_t}$  and re-normalize  $T_{A_t}$  when calculating  $\hat{T}_{U_t}$ , so that the user simulator will less likely talk about this topic in its response  $U_t$ . We denote this procedure as the *interest decay mechanism*. In our experiments, we use a *hard interest decay mechanism*, where the proportion of the major topic is set to 0 in  $T_{A_t}$  when the decay mechanism is triggered.

Finally, we allow the user simulator to terminate the conversation upon receiving dull or improper agent responses. Three types of agent responses are considered: (1) Dull responses, e.g., “I’m not sure what you are talking about”. We follow Li et al. (2016b) to detect dull responses with a string matching method, i.e., by comparing the response with a set of manually written templates. Specifically, we consider totally 27 phrases, such as “I don’t know what you mean” and “I don’t know if it’s a joke”, as dull phrases. Any response containing one of the phrases in its string is considered as a dull response. (2) Highly overlapping consecutive responses (under exact word matching). We define two consecutive responses to be highly overlapping when there are more than 60% word overlaps among them. (3) Duplicate responses, i.e., when the agent has provided the same response three times in the conversation history.

#### 3.2 Topic Detector

In our user simulator shown in Figure 2, the *Topic Detector* (TD) is used to detect the topic  $T_{A_t}$  of the agent utterance  $A_t$ . We adopt SWEM-concat (Shen et al. 2018) as our topic detector. Intrinsically, given an utterance, the SWEM-concat model first computes two kinds of utterance representations, by averaging and max pooling over the embedding vectors of words in this utterance. It then concatenates the two representation vectors to form a final utterance representation vector. This vector is further fed into a two-layer ReLU-based fully connected neural network to make a prediction. We train the topic detector by minimizing its cross entropy loss compared with the ground-truth topic label, using supervised data.

#### 3.3 Topic-Aware Response Generator

We employ a *Topic-Aware Response Generator* (TA-RG) to generate user responses in the simulator, which conceptually relates to the persona model proposed by Li et al. (2016a). Essentially, given an agent utterance  $A_t = \{a_1, a_2, \dots, a_{|A_t|}\}$  and a target topic distribution  $\hat{T}_{U_t}$ , the goal of TA-RG is to produce a response  $U_t = \{u_1, u_2, \dots, </s>\}$ , where  $</s>$  is a special token indicating the end of the sentence, such that the response can reflect the endowed target topic distribution  $\hat{T}_{U_t}$ .

Inspired by Li et al. (2016a), we structure TA-RG with a *conditional Encoder-Decoder* model. As shown in Figure 2, the agent utterance  $A_t$  is encoded via a bi-directional LSTM neural network (Hochreiter and Schmidhuber 1997), i.e.,

$$\begin{aligned} \vec{h}_i^A &= \text{LSTM}^{enc}(W_e a_i, \vec{h}_{i-1}^A), i \in [1, |A_t|], \\ \overleftarrow{h}_i^A &= \text{LSTM}^{enc}(W_e a_i, \overleftarrow{h}_{i+1}^A), i \in [|A_t|, 1], \\ h^A &= [\vec{h}_{|A_t|}^A; \overleftarrow{h}_1^A], \end{aligned}$$

where  $W_e \in \mathbb{R}^{d_e \times |V|}$  is a trainable word embedding matrix,  $|V|$  is the word vocabulary size, and the final vector representation of  $A_t$ , i.e.,  $h^A \in \mathbb{R}^d$ , is defined as the concatenation of the last hidden states  $\vec{h}_{|A_t|}^A$  and  $\overleftarrow{h}_1^A$  in both directions.

For the decoder, we use another LSTM-based recurrent neural network, conditioned on the given topic  $\hat{T}_{U_t}$ , to generate a *topic-aware* response. Specifically, we use a trainable topic embedding matrix  $W_T \in \mathbb{R}^{d_T \times K}$  to encode each topic distribution. The input to the decoder LSTM thus becomes the concatenation of the embedding vectors of the word  $u_j$  and the topic  $\hat{T}_{U_t}$ , i.e.,

$$h_j^U = \text{LSTM}^{dec}([W_e u_j; W_T \hat{T}_{U_t}], h_{j-1}^U), j \geq 1,$$

where  $h_0^U = h^A$ . To generate the next word  $u_{j+1}$ , an attention mechanism following (Luong, Pham, and Manning 2015) is applied, where a matrix  $W_c \in \mathbb{R}^{d \times 2d}$  is learned to summarize the decoder hidden state  $h_j^U$  and the attended encoder hidden states  $h_i^A$  into  $\tilde{h}_j^U$ :

$$\alpha_{ij} = \frac{\exp((h_j^U)^\top h_i^A)}{\sum_{i'} \exp((h_j^U)^\top h_{i'}^A)}, h_i^A = [\overset{\rightarrow}{h}_i^A; \overset{\leftarrow}{h}_i^A],$$

$$c_j = \sum_{i=1}^{|A_t|} \alpha_{ij} h_i^A, \tilde{h}_j^U = \tanh(W_c [c_j; h_j^U]).$$

The probability of the next word is given by

$$P(u_{j+1}|u_{\leq j}, A_t, \hat{T}_{U_t}) = \text{softmax}(W \tilde{h}_j^U + b),$$

where  $W \in \mathbb{R}^{|V| \times d}$ ,  $b \in \mathbb{R}^{|V|}$ . The log-likelihood objective can be written as  $\mathcal{L}_s = \sum_{j=1}^{|U_t|} \log P(u_{j+1}|u_{\leq j}, A_t, \hat{T}_{U_t})$ . We use Beam Search to generate multiple response candidates, where a sibling penalty (Li, Monroe, and Jurafsky 2016) is applied to promote diversity among words at each decoding step, and a length normalization factor is used to avoid favoring short sentences (Wu et al. 2016). We train TA-RG in the teacher-forcing manner, using a dataset of source-target utterance pairs, with their true topic given.

**Response Reranking** To encourage responses with closer topic distributions to the target topic  $\hat{T}_{U_t}$ , for each generated candidate  $U_t$ , we evaluate the Euclidean distance  $\mathcal{L}_c = \|T_{U_t} - \hat{T}_{U_t}\|$  between the predicted topic of  $U_t$  (i.e.,  $T_{U_t}$ ) and  $\hat{T}_{U_t}$ . This is conceptually related to (Hu et al. 2017) in that the generated response is encouraged to reflect the input topical signal  $\hat{T}_{U_t}$ . On the other hand, to discourage responses with repetitive patterns, we apply a penalty  $\mathcal{L}_r = 0.1 * C$  for each response, where  $C$  is the count of the most frequent word (except stop words) in this response. We finally rerank response candidates in the increasing order of  $\mathcal{L}_c + \mathcal{L}_r$ , and randomly select a response from the top-10 as  $U_t$ .

## 4 Baseline Conversational Agents

In general, agents initiate conversations with no prior information about the user interest distribution  $T_U$ . The objective is to infer  $T_U$  through the conversation. Such agents have four principal components: (1) The topic detector (TD) module, which detects the topic of the user utterance  $U_t$ . This is the same TD as in Section 3.2. (2) The decision-making (DM) module, which makes a strategic decision to select the next topic for generating a response. (3) The response generation (RG) module, which synthesizes a response to elicit

DM	RG	
	Greedy + template Follow + template Neural + template	Greedy + TA-RG Follow + TA-RG Neural + TA-RG

Table 1: Tested agents in our experiments. DM: Decision-Making module. RG: Response Generator.

the user’s interest. (4) The interest prediction (IP) module that infers  $T_U$ .

As shown in Figure 2, after receiving a user utterance  $U_t$ , the agent first detects its topic using the TD module. The DM module takes the resulting *predicted user topic*  $T_{U_t}$  (and potentially all previously predicted user topics  $T_{U_{<t}}$ ) as inputs, to produce a *target agent topic distribution*  $D_{t+1} \in \mathbb{R}^K$ . This is followed by the RG module, which uses  $D_{t+1}$  and  $U_t$  to generate an agent utterance  $A_{t+1}$ . The final module IP predicts the user interest distribution  $T_U$  by averaging over all  $D_t$  across the conversation.

In this paper, we investigate 6 agents in total, which share the same TD and IP modules but use 2 different RG modules (i.e., template-based RG and TA-RG) and 3 different DM modules (i.e., “Greedy”, “Follow” and a neural decision network). Table 1 summarizes the compared agents in our study.

### 4.1 User Interest Prediction Module

The agent makes a prediction of the user’s interest at the end of the conversation (i.e., when the user leaves the chat or the conversation has proceeded for maximum turns). We average<sup>1</sup> over all the agent’s decisions to estimate the user’s interest  $T^{pred}$  (in a distribution), i.e.,

$$T^{pred} = \frac{\sum_{t=2}^T D_t}{T-1}, \quad (2)$$

where  $T$  is the actual number of turns in this conversation. The first decision  $D_1$  is randomly picked (when the agent initiates the conversation) and thus is not counted. The final topic prediction is the most likely one (i.e.,  $\text{argmax}_{i \in [1, K]} (T^{pred})_i$ ).

### 4.2 Decision-Making Module

The DM module takes the user utterance  $U_t$  (and potentially the dialogue history) to decide the next conversation topic for eliciting the user’s interest. As shown in Table 1, we explore three decision strategies for the DM module: the naïve Greedy and Follow strategies, and a neural decision network, as follows.

**Greedy** The Greedy strategy takes the major topic (i.e., the most prominent topic) in  $T_{U_t}$  as the next one, i.e.,  $D_{t+1} = \mathbb{1}(\text{argmax}_{i \in [1, K]} (T_{U_t})_i)$ , where  $\mathbb{1}(i)$  is a one-hot vector with 1 at the  $i$ -th position.

<sup>1</sup>We take an average such that a successful agent has to spend most of its topics on the user’s interest. While there are also other viable formulations (e.g., taking the last decision  $D_T$  as the prediction), they may make the training more difficult (e.g., requiring reinforcement learning), thus we leave them for future studies.

**Follow** The Follow strategy copies the predicted user topic as the target agent topic distribution, i.e.,  $D_{t+1} = T_{U_t}$ .

**Neural Decision Network** We further propose a *Neural Decision Network* as a DM module, attempting to *proactively* drive the conversation. The architecture of this network is shown in Figure 2. This strategy aims to decide the next topic  $D_{t+1}$  based on the currently predicted user topic  $T_{U_t}$  and the previously predicted user topics  $T_{U_{<t}} = \{T_{U_1}, \dots, T_{U_{t-1}}\}$ . Particularly, by leveraging the previously predicted user topics, we hope the decision network can capture every nuance of the user topic throughout the conversation. For example, if a user shows increasing interest in certain topics, these specific topics are likely to contain their actual interest.

In our experiments, we utilize a simplified design of this decision network by taking the user’s current topic  $T_{U_t}$  as well as the difference between their previous topics (i.e.,  $T_{U_t} - T_{U_{t-1}}$  and  $T_{U_{t-1}} - T_{U_{t-2}}$ ) as inputs. A fully-connected layer with  $\tanh$  as the activation function and a linear layer are applied to the input vectors. Specifically,

$$\begin{aligned} s_t &= [T_{U_t}; T_{U_t} - T_{U_{t-1}}; T_{U_{t-1}} - T_{U_{t-2}}], \\ o &= \tanh(W_1 s_t + b_1), \\ D_{t+1} &= \text{softmax}(W_2 o + b_2), \end{aligned}$$

where  $W_1 \in \mathbb{R}^{3K \times 3K}$ ,  $b_1 \in \mathbb{R}^{3K}$ ,  $W_2 \in \mathbb{R}^{K \times 3K}$  and  $b_2 \in \mathbb{R}^K$ , are trainable model parameters.

The decision network is trained to minimize the cross-entropy loss  $\mathcal{L}$  between the true user interest  $T_U$  and the final prediction  $T^{pred}$  in (2), i.e.,

$$\mathcal{L} = - \sum_{i=1}^K (T_U)_i \log (T^{pred})_i. \quad (3)$$

In practice, to accelerate training of the decision network, we first pretrain it as an auto-encoder to imitate the Follow strategy. Specifically, for each training example in a mini-batch, we simulate the training input  $T_{U_{t,n}}$  by sampling a set of  $K$ -dim probability distributions from a Dirichlet distribution. The decision network is then pretrained to minimize the Mean Square Error between the predicted distribution  $D_{t+1,n}$  and the input distribution  $T_{U_{t,n}}$ , i.e.,  $\frac{1}{N} \sum_{n=1}^N \|D_{t+1,n} - T_{U_{t,n}}\|^2$ , where  $N$  is the batch size.

The subsequent training procedure is described as follows: for each iteration, we simulate one dialogue with the current agent until it ends. The loss in (3) is then calculated, and the gradients are back-propagated to the parameters in the decision network. Note that the only trainable parameters in this procedure are in the decision network, while other parts (e.g., the response generator and topic detector module) are fixed.

### 4.3 Response Generation Module

We consider two approaches to generate the next agent utterance  $A_{t+1}$ : the neural topic-aware generator (TA-RG) and the template-based RG.

Topic	NFL	NBA	Soccer	Bicycling	Golf
Size	40K	40K	40K	40K	41,156
Topic	Boxing	Electronic music	Classical music	Comic books	Fantasy novel
Size	21,576	20,897	9,032	40K	28,767

Table 2: Number of conversations per topic in our data.

**TA-RG** The TA-RG for agent utterance generation shares the same structure as in the user simulator response generation module. Given the user utterance  $U_t$  and the target agent topic distribution  $D_{t+1}$ , the TA-RG model produces a topic-aware response as described in Section 3.3. The first target topic  $D_1$  is a random one-hot vector.

**Template-based RG** To better understand each decision strategy and as a baseline to demonstrate the effectiveness of TA-RG, we also consider template-based response generation. For template-based response generation, we manually write two templates for each topic candidate (shown in Appendix 9). Given the target agent topic distribution  $D_t$  (including the random topic  $D_1$  selected in the beginning of the conversation), we retrieve one template of its most prominent topic candidate (indexed by  $\arg\max_{i \in [1, K]} (D_t)_i$ ) as the response  $A_t$ . Simple rules, such as not selecting the same template in consecutive two turns for a topic, are also designed to avoid naive conversation termination (Section 3.1).

Note that, with the template-based RG, the response for the Greedy strategy and the Follow strategy would be the same (given the same  $D_t$ ), as they have the same  $\arg\max$  value in  $D_t$ . However, the two strategies can have different user interest estimations from IP based on (2), because their  $D_t$  distributions may be different.

## 5 System Evaluation

Each agent is evaluated by its success rate, i.e., the average user interest prediction accuracy, over 500 conversations. For each conversation, a topic candidate is randomly picked as the true user interest, and a prediction is considered correct when  $\mathbb{1}(\arg\max_{i \in [1, K]} (T^{pred})_i) = T_U$ , where  $T^{pred}$  is the user interest estimation in (2) and  $T_U$  is the true user interest. We repeat the testing for three times with different random seeds and report the average performance as well as its standard deviation. For systems equipped with a neural response generator, we also evaluate the relevance and diversity of generated responses based on automatic metrics.

## 6 Experiments

### 6.1 Dataset

We train our topic-aware response generator and the topic detector on the **Reddit dataset** (Zhang et al. 2018b; Yoshino et al. 2019), which contains more than 300,000 multi-turn conversations collected from 10 “subreddit” divisions, covering 10 topics in Sports, Music and Books. The data statistics are shown in Table 2. Each utterance was Treebank-tokenized and lowercased.

To train the topic-aware response generator, we reorganize conversations for each topic into triples of  $(T_U, \text{source}, \text{target})$ , where  $T_U$  is a one-hot vector of the



<b>Topic:</b> Boxing
<b>Input:</b> “I usually watch sport games at home”
<b>Responses:</b> have you watched the <i>fight</i> s? rooting for <i>pascal</i> . <i>mayweather</i> vs <i>pascal</i> ? gets <i>knocked down</i> . matches in the <i>nosebleeds</i> .
<b>Topic:</b> Comic books
<b>Input:</b> “The marvel new season is released!”
<b>Responses:</b> also <i>rotworld</i> is this the new <i>remender</i> series? ultimate <i>spider-man</i> isn’t released yet. but the new <i>avengers</i> isn’t released yet. <i>captain marvel</i> is officially out of print.

Table 3: Responses generated by our topic-aware response generator. Words in *italic* are topic-related.

Generator	BLEU-4	Dist-1	Dist-2	Ent-4
Human	N/A*	0.148	<b>0.632</b>	<b>9.961</b>
Random	0.161	0.148	<b>0.632</b>	<b>9.961</b>
Templates	0.000	0.001	0.001	1.029
TA-RG	0.974	<b>0.163</b>	0.394	7.114

Table 4: Results for TA-RG vs. ground-truths, test set random permutation, and template-based RG. \*Human BLEU can only be measured with at least two references per input as in (Li et al. 2016a, Table 2), which we don’t have.

topic candidate. The average lengths of the sources and the targets are 28 and 14, respectively. We split the triples for training (317,440 triples), validation (1,994 triples) and testing (1,994 triples). To train the topic detector, we simply pair each single utterance with its ground-truth topic label  $T_U$  as training data. The training/validation/testing split is exactly the same as the aforementioned. To deal with imbalanced data size of different topics (verified to be useful in our experiment), we perform down sampling for frequent topics and up sampling for rare topics, resulting in around 30K instances for each topic.

## 6.2 Model Setup

**General Setup** For each conversation, the maximum number of turns is set to 30, i.e., at most 15 utterances for both the user and the agent. We implement all our framework with TensorFlow.<sup>2</sup>

**Topic Detector** For TD, we use the same hyper-parameter setting as (Shen et al. 2018). Specifically, the word embedding size is 300, and the dimension of the fully connected layer is also 300. The dropout rate is set to 0.5. The model is updated via ADAM (Kingma and Ba 2015) with batch size as 128 and learning rate as  $3e-4$ .

**Topic-Aware Response Generator** For TA-RG, we set the word embedding size  $d_e = 300$  (initialized by Glove (Pennington, Socher, and Manning 2014)) and the topic embed-

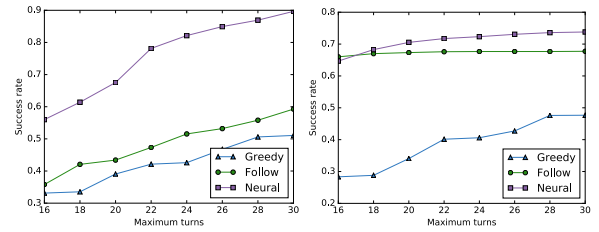


Figure 3: Success rates of template-based (left) and TA-RG-based (right) agents in response to different maximum conversation turns.

ding size  $d_T = 512$ . The  $LSTM^{enc}$  and  $LSTM^{dec}$  unit sizes are 256 and 512, respectively (i.e.,  $d = 512$ ). Parameters are updated via ADAM (Kingma and Ba 2015) with batch size 128 and learning rate 0.001. The vocabulary size is set to  $|V| = 20,001$  by selecting most frequent words. When generating utterances, the beam size is set to 500, the maximum decoding length is 30, the sibling penalty weight is 10, and the length normalization factor is 0.5.

**User Simulator** We set  $\lambda = 0.3$  when calculating  $\hat{T}_{U_t}$  in (1). For the “interest decay” mechanism, we set  $S$  to 3 and use hard interest decay, as described in Section 3.

**Conversational Agents** We implement all agents with the same TD and TA-RG (if used) as in the user simulator. The neural decision network, after being pretrained, is further trained for 1,000 iterations (i.e., conversations), as explained in Section 4.2.

## 6.3 Results

**Comparisons on generation modules** For qualitative evaluation, we present responses generated by our TA-RG in Table 3. For quantitative evaluation, we follow Zhang et al. (2018b) to evaluate the relevance and diversity of the generation by measuring BLEU, Dist-1/2 and Ent-4 scores (Table 4). We observed that TA-RG can produce much more diverse responses comparing to template-based RG and its relevance (evaluated by BLEU) significantly outperforms randomly permuted source-target pairs on test set.

**Comparisons on agent decision modules** The results of various decision modules are shown in Table 5. Whenever equipped with the template-based RG or the TA-RG, the neural decision strategy always achieves the best success rate in identifying user interests. We presume that the Greedy and the Follow strategy either *passively* repeat the major topic of the last user utterance or *passively* copy the predicted user topic. Examples are shown in Appendix 10. When the maximum turn is reached, the user may leave the conversation before revealing the real interest to the agent. Under such a scenario the two naïve agents may not be able to learn what the user is interested in. This is akin to the real-life situation where customers would hide their actual thoughts before establishing the trust in the salesman.

Our proposed neural decision network, however, is trained to optimize the success rate, thus is more likely to “proactively” drive the conversation. In most cases, the neural decision agents can precisely capture the user’s subtle increase in

<sup>2</sup><https://www.tensorflow.org/>.



Figure 4: An example conversation between the simulated user, who is interested in NFL, and our TA-RG-based agent (with the neural decision network). Our agent can capture changes in the user topic and successfully suggests the correct topic ( $D_4$ ) before the user shows an obvious preference. Not involved topics are omitted for simplicity.

Agent	Template	TA-RG
Greedy	0.510 ( $\pm 0.008$ )	0.477 ( $\pm 0.012$ )
Follow	0.583 ( $\pm 0.012$ )	0.677 ( $\pm 0.003$ )
Neural Decision	<b>0.902 (<math>\pm 0.006</math>)</b>	<b>0.738 (<math>\pm 0.021</math>)</b>

Table 5: Success rates of agents with different decision strategies and response generators. Numbers within the parentheses are standard deviations over three runs.

their topic of interest and decrease in other topics, and push the dialogue flow towards the user’s real interest by strengthening the weight of the interesting topic in  $D_{t+1}$ . Take the “Neural Decision+Template” agent for example. In 260 out of the 500 testing conversations, the agent can successfully suggest a correct topic (i.e.,  $\mathbb{1}(\arg\max_i(D_{t+1})_i) = T_U$ ) even before the user shows an obvious preference (i.e.,  $\mathbb{1}(\arg\max_i(T_{U_t})_i) \neq T_U, \forall t \geq 1$ ). With such behaviors, the IEC agent, when equipped with a neural decision network, is more likely to elicit the user’s interest at an early stage as illustrated in Figure 3, which compares each agent’s success rate against different maximum conversation turns.

For agents equipped with the template-based RG (where a set of predefined agent utterances are given), we observe that the decision network may trivially “memorize” the user response topic ( $T_{U_t}$ ), leading to a high success rate of 90%. For the TA-RG agent, however, the freely generated utterances can not be enumerated, rendering it much more challenging for the decision network to optimize its strategy. Nonetheless, we still observe that the decision network is capable to learn to first follow the topic then aggressively select the topic that might not be most prominent one but demonstrates high persistence over the entire dialog. This can be illustrated by the example in Figure 4. Our agent begins with a randomly picked topic “soccer”. After three turns, the agent detects continuously increased user’s interest in “NFL” and decreased interest in “soccer”, and thus

decides to adjust the conversation topic towards the former at the 4-th turn, with a larger weight  $(D_4)_1 = 0.5$ . When the user echoes this preference with  $(T_{U_4})_1 = 0.83$ , the agent further strengthens it (i.e.,  $(D_5)_1 = 0.96$ ). The theme of the subsequent conversation utterances is all about the user’s interest “NFL”, and the agent finally correctly predicts the user’s interest as “NFL”.

## 7 Conclusion and Future Work

In this work, we have taken the first steps towards building a conversational agent that elicits users’ interests in conversations. A simulated environment and several competitive agents are provided for future research. These promising results suggest several directions for future exploration. First, more complex and realistic human behaviors will be encoded into the environment to construct a more practical IEC bot. Second, to further facilitate this line of research, we will integrate interactive human evaluation into the framework, something that is desirable, indeed essential, for both practical and scientific reasons. Third, this framework has the potential to function as a useful conversation testbed for reinforcement learning to optimize both the decision-making module and the generator in order to maximize the success rate, in a manner analogous with (Williams and Young 2007; Li et al. 2016b). In addition, we envisage simultaneously optimizing the agent towards eliciting user interest (multi-turn goal), and improving aspects of response quality such as diversity and relevance (single-turn goal).

## References

- Das, A.; Kottur, S.; Moura, J. M.; Lee, S.; and Batra, D. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *ICCV*.
- De Vries, H.; Strub, F.; Chandar, S.; Pietquin, O.; Larochelle, H.; and Courville, A. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *CVPR*.

- El Asri, L.; He, J.; and Suleman, K. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. *Interspeech 2016*.
- Fang, H.; Cheng, H.; Sap, M.; Clark, E.; Holtzman, A.; Choi, Y.; Smith, N. A.; and Ostendorf, M. 2018. Sounding board: A user-centric and content-driven social chatbot. In *NAACL: Demonstrations*.
- Gao, J.; Galley, M.; Li, L.; et al. 2019. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval* (2-3).
- Glas, N., and Pelachaud, C. 2015. User engagement and preferences in information-giving chat with virtual agents. In *Workshop on Engagement in Social Intelligent Virtual Agents*.
- Glas, N.; Prepin, K.; and Pelachaud, C. 2015. Engagement driven topic selection for an information-giving agent. In *Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2015-goDial)*.
- He, H.; Chen, D.; Balakrishnan, A.; and Liang, P. 2018. Decoupling strategy and generation in negotiation dialogues. In *EMNLP*.
- Henderson, M.; Thomson, B.; and Williams, J. D. 2014. The second dialog state tracking challenge. In *SIGDIAL*.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* (8).
- Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*.
- Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- Kreyszig, F.; Casanueva, I.; Budzianowski, P.; and Gasic, M. 2018. Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *SIGDIAL*.
- Lewis, M.; Yarats, D.; Dauphin, Y.; Parikh, D.; and Batra, D. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *EMNLP*.
- Li, J.; Galley, M.; Brockett, C.; Spithourakis, G.; Gao, J.; and Dolan, B. 2016a. A persona-based neural conversation model. In *ACL*.
- Li, J.; Monroe, W.; Ritter, A.; Jurafsky, D.; Galley, M.; and Gao, J. 2016b. Deep reinforcement learning for dialogue generation. In *EMNLP*.
- Li, X.; Lipton, Z. C.; Dhingra, B.; Li, L.; Gao, J.; and Chen, Y.-N. 2016c. A user simulator for task-completion dialogues. *arXiv*.
- Li, X.; Panda, S.; Liu, J.; and Gao, J. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv*.
- Li, J.; Monroe, W.; and Jurafsky, D. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv*.
- Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.
- Mazare, P.-E.; Humeau, S.; Raison, M.; and Bordes, A. 2018. Training millions of personalized dialogue agents. In *EMNLP*.
- Misu, T., and Kawahara, T. 2007. An interactive framework for document retrieval and presentation with question-answering function in restricted domain. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer.
- Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP*.
- Qin, J.; Ye, Z.; Tang, J.; and Liang, X. 2020. Dynamic knowledge routing network for target-guided open-domain conversation. AAAI Press.
- Ram, A.; Prasad, R.; Khatri, C.; Venkatesh, A.; Gabriel, R.; Liu, Q.; Nunn, J.; Hedayatnia, B.; Cheng, M.; Nagar, A.; et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv*.
- Schurr, P. H., and Ozanne, J. L. 1985. Influences on exchange processes: Buyers' preconceptions of a seller's trustworthiness and bargaining toughness. *Journal of consumer research* (4).
- Shen, D.; Wang, G.; Wang, W.; Renqiang Min, M.; Su, Q.; Zhang, Y.; Li, C.; Henao, R.; and Carin, L. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *ACL*.
- Sordoni, A.; Galley, M.; Auli, M.; Brockett, C.; Ji, Y.; Mitchell, M.; Nie, J.-Y.; Gao, J.; and Dolan, B. 2015. A neural network approach to context-sensitive generation of conversational responses. In *NAACL*.
- Swan, J. E., and Nolan, J. J. 1985. Gaining customer trust: a conceptual guide for the salesperson. *Journal of Personal Selling & Sales Management* (2).
- Tang, J.; Zhao, T.; Xiong, C.; Liang, X.; Xing, E.; and Hu, Z. 2019. Target-guided open-domain conversation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5624–5634.
- Vinyals, O., and Le, Q. 2015. A neural conversational model. *arXiv*.
- Wen, T.-H.; Vandyke, D.; Mrkšić, N.; Gasic, M.; Barahona, L. M. R.; Su, P.-H.; Ultes, S.; and Young, S. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*.
- Williams, J. D., and Young, S. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language* (2).
- Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv*.
- Wu, W.; Guo, Z.; Zhou, X.; Wu, H.; Zhang, X.; Lian, R.; and Wang, H. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3794–3804.
- Yoshino, K., and Kawahara, T. 2015. Conversational system for information navigation based on pomdp with user focus tracking. *Computer Speech & Language* (1).
- Yoshino, K.; Hori, C.; Perez, J.; D'Haro, L. F.; Polymenakos, L.; Gunasekara, C.; Lasecki, W. S.; Kummerfeld, J. K.; Galley, M.; Brockett, C.; et al. 2019. Dialog system technology challenge 7. *arXiv*.
- Yu, Z.; Black, A. W.; and Rudnicky, A. I. 2017. Learning conversational systems that interleave task and non-task content. In *IJCAI*. AAAI Press.
- Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; and Weston, J. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? In *ACL*.
- Zhang, Y.; Galley, M.; Gao, J.; Gan, Z.; Li, X.; Brockett, C.; and Dolan, B. 2018b. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*.
- Zhou, L.; Gao, J.; Li, D.; and Shum, H.-Y. 2018. The design and implementation of xiaoice, an empathetic social chatbot. *arXiv*.



## 8 Details for User Behavior Simulation

We model the user losing interest when the agent has discussed a less interesting topic several times. We consider the most prominent topic (indexed by  $\text{argmax}_{i \in [1, K]} (T_{A_t})_i$ ) as the *major topic* of an agent utterance  $A_t$ . If a topic (different from the user’s interest) has been the major topic for  $S$  times (in our experiments  $S = 3$ ), we decay its value in  $T_{A_t}$  and re-normalize  $T_{A_t}$  when calculating  $\hat{T}_{U_t}$ , so that the user simulator will less likely talk about this topic in its response  $U_t$ . We denote this procedure as the *interest decay* mechanism. In our experiments, we use a *hard interest decay* mechanism, where the proportion of the major topic is set to 0 in  $T_{A_t}$  when the decay mechanism is triggered.

We endow the user simulator the ability to terminate the conversation upon receiving dull or improper agent responses. Three types of agent responses can cause termination: (1) Dull responses, e.g., “I’m not sure what you are talking about”. We follow Li et al. (2016b) to detect dull responses with a string matching method, i.e., by comparing the response with a set of manually written templates. Specifically, we consider totally 27 phrases, such as “I don’t know what you mean” and “I don’t know if it’s a joke”, as dull phrases. Any response containing one of the phrases in its string is considered as a dull response. (2) Highly overlapping consecutive responses (under exact word matching). We define two consecutive responses to be highly overlapping when there are more than 60% word overlaps among them. (3) Duplicate responses, i.e., when the agent has provided the same response three times in the conversation history.

## 9 Template-based Response Generator

As introduced in Section 4.3, we prepared two template responses for each of the 10 topics, as shown in Table 6.

Topic	Templates
NFL	which nfl team do you support who’s your favourite nfl player
NBA	are you a nba fan who was the mvp last year
Soccer	i watch european cup when i’m free i’m a fan of real madrid
Bicycling	i ride for 10 miles every weekend it is fun to ride a bike
Golf	there’s a new golf club nearby are you a fan of tiger woods
Boxing	i like watching boxing fights what fight have you watched this week
Electronic music	i like dancing with dj music do you listen to daft punk
Classical music	do you play instruments? can you play the piano
Comic books	the marvel new season is released! batman or superman?
Fantasy novel	what fantasy novel have you read which fiction writer do you like

Table 6: Templates used by template-based agents for each topic.

## 10 Examples of Greedy/Follow Strategies

We show examples in Table 7 and Table 8 for TA-RG-based agents adopting Greedy or Follow strategy, respectively. For both of the naive strategies, the agents tend to passively “echo” the user’s last utterance, making them easy to fall into a meaningless chatting cycle (e.g., discussing about different new issued comics in Table 7). In comparison, the agent equipped with a neural decision module can drive the conversation and is less likely to fall into such cycles, as shown in Figure 4.

---

### User interest (unknown to agent): NBA

---

Agent: the marvel new season is released!

User: up for the next issue of the year.

Agent: hickman’s run is a great issue.

User: one of the best in the league is a great issue.

Agent: deadpool is a great issue.

**User leaves the conversation!**

**Agent conclusion: The user likes comics books.**

---

Table 7: Conversation example for the TA-RG-based agent with the Greedy decision strategy. The agent passively echoed the user’s utterance and made a wrong prediction in the end.

---

### User interest (unknown to agent): NBA

---

Agent: the marvel new season is released!

User: up for the next issue of the year.

Agent: up for the next issue of the year.

User: 2nd issue of the year.

Agent: question: what is the 2nd issue?

User: try the 2nd issue.

Agent: twist: the second issue is the 2nd.

**User leaves the conversation!**

**Agent conclusion: The user likes comics books.**

---

Table 8: Conversation example for the TA-RG-based agent with the Follow decision strategy. The agent passively echoed the user’s utterance and made a wrong prediction in the end.