Learning to Select Knowledge for Response Generation in Dialog Systems

Rongzhong Lian[†], Min Xie[‡], Fan Wang[†], Jinhua Peng[†], Hua Wu[†]

[†] Baidu Inc. China

[‡] The Hong Kong University of Science and Technology

{lianrongzhong, wangfan04, pengjinhua, wu_hua}@baidu.com

mxieaa@cse.ust.hk

Abstract

Generating informative responses in end-to-end neural dialogue systems attracts a lot of attention in recent years. Various previous work leverages external knowledge and the dialogue contexts to generate such responses. Nevertheless, few has demonstrated their capability on incorporating the appropriate knowledge in response generation. Motivated by this, we propose a novel open-domain conversation generation model in this paper, which employs the posterior knowledge distribution to guide knowledge selection, therefore generating more appropriate and informative responses in conversations. To the best of our knowledge, we are the first one who utilize the posterior knowledge distribution to facilitate conversation generation. Our experiments on both automatic and human evaluation clearly verify the superior performance of our model over the state-of-the-art baselines.

1 Introduction

Over the past few years, open-domain dialogue systems are attracting much attention of researchers owing to their great potential in real applications such as educational robot, emotional companion and chitchat. The recently proposed sequence-to-sequence (Seq2Seq) model (Shang et al., 2015; Vinyals and Le, 2015; Cho et al., 2014b) has garnered considerable attention due to its simplicity and wide applicability. Despite its popularity, the traditional Seq2Seq models tend to produce generic and less informative responses, such as *"I don't know"* and *"That's cool"*, resulting in less attractive conversations.

To address this limitation, a variety of neural models (Zhou et al., 2018; Ghazvininejad et al.,

---

Work done when Min Xie was an intern at Baidu Inc.

Source	Hi! I do not have a favorite band but my favorite reading is twilight.
Profiles/ Knowledge	K1. I love the band red hot chili peppers.
	K2. My feet are size six women s.
	K3. I want to be a journalist but instead i sell washers at sears.
	K4. I have a french bulldog.
R1 (no knowledge)	What do you do for a living?
R2 (use K2)	I bought a pair of shoes of size six women.
R3 (use K3)	I am a good journalist.
R4 (use K3)	I also like reading and wish to be a journalist, but now can only sell washers.
Target	I love to write! want to be journalist but have settle for selling washers at sears.

Table 1: Comparison between Various Responses Generated with/without Correct Knowledge

2018; Liu et al., 2018) were proposed to leverage external knowledge (which can be either unstructured texts or structured triples), aiming at producing more appropriate and informative responses. For example, the commonsense model proposed in (Zhou et al., 2018) takes commonsense knowledge into account, which is served as knowledge background to facilitate conversation understanding. The recently created datasets Persona-chat (Zhang et al., 2018) and Wizard-of-Wikipedia (Dinan et al., 2018) introduce conversation-related knowledge (e.g., the personal profiles in Persona-chat) in response generation where knowledge is used to direct the conversation flow. Dinan et al. (2018) used ground-truth knowledge to guide knowledge selection, which demonstrates improvements over those not using such information. However, most of the existing researches only make use of the semantic similarity of the input utterance and the knowledge, which are incorporated into the decoder for response generation, without effective mechanisms to guarantee the appropriate knowledge selection and generation in responses.

The problems are further illustrated in Table

1, which is a dialogue from Persona-chat ([Zhang et al., 2018](#)). In this dataset, each agent is associated with a persona profile, which is served as knowledge. Two agents exchange persona information based on the associated knowledge. Given the source utterance, different responses can be generated depending on whether appropriate knowledge is properly used. For example, in R1, no knowledge is leveraged and thus, it produces a safe but less informative response. R2, which incorporates the wrong knowledge K2, is an improper response. The appropriate knowledge K3 is used in both R3 and R4. However, R3 is a less relevant response since it fails to properly leverage K3 in generating responses while R4 is a meaningful response since K3 is fully utilized in it. The choice and the incorporation of knowledge plays a vital role in generating appropriate responses, which, however, is underexplored in existing models. Apart from this, we observe that since the selected knowledge K3 occurs in the response, we wonder if the target response can be effectively utilized so as to provide directive information on choosing and incorporating knowledge.

To overcome the aforementioned deficiencies, we propose to use both input utterances and response (instead of only input utterances) to guide knowledge selection and incorporation in open-domain response generation. To achieve this, we define a posterior probability distribution over knowledge conditioned on both input utterances and response during the training phase. With the help of this posterior distribution, the model tends to select appropriate knowledge for response generation. Unfortunately, the target response is only available in training. Thus, when generating responses (i.e., the target response is unknown), another *prior knowledge distribution* is proposed to accurately approximate the posterior knowledge distribution. For this purpose, we train the prior knowledge distribution using the posterior knowledge distribution as a guidance. Then, augmented with the knowledge distributions, our model is able to select appropriate knowledge and to generate proper and informative responses.

Our contributions are shown as follows:

- We are the first to propose a novel conversation model where *posterior knowledge* is utilized to enable both effective knowledge selection and incorporation. Under the effective guidance of posterior knowledge distri-bution, our model is capable to choose appropriate knowledge and to generate informative responses even when the posterior knowledge is not available.

- Comprehensive experiments demonstrate that our model significantly outperforms the existing ones by incorporating knowledge more properly and generating more appropriate and informative responses.

2 Model

In this paper, we focus on the problem of grounding a neural model with knowledge selection and incorporation. Formally, given a source utterance $X = x_1 x_2 \ldots x_n$ (where x_t is the t-th word in X) and a collection of knowledge $\{K_1, \ldots, K_N\}$ (which can be either unstructured texts or structure triples), the goal is to select appropriate knowledge from the knowledge collection and then, to generate a target response $Y = y_1 y_2 \ldots y_m$ by incorporating the selected knowledge properly.

2.1 Background: Seq2Seq and Attention

Before we go into the model details, we provide a brief introduction on the required technical background, namely the Seq2Seq model and the attention mechanism. The readers who are familiar with these concepts can simply skip this section.

Seq2Seq. Seq2Seq ([Vinyals and Le, 2015](#)) follows a typical encoder-decoder framework.

Given a source utterance $X = x_1 \ldots x_n$, the encoder encodes it into a sequence of hidden states:

$$\mathbf{h}_t = f(x_t, \mathbf{h}_{t-1}) \tag{1}$$

where $\mathbf{h}_t$ is the hidden state of the encoder at time t and f is a non-linear transformation, which can be a long-short term memory unit (LSTM) ([Hochreiter and Schmidhuber, 1997](#)) or a gated recurrent unit (GRU) ([Cho et al., 2014a](#)). In particular, the last hidden state $\mathbf{h}_n$ is also used as the context vector of X and it is denoted by $\mathbf{c} = \mathbf{h}_n$.

Then, the decoder generates a target utterance $Y = y_1 y_2 \ldots y_m$ sequentially conditioned on the given context vector $\mathbf{c}$. Specifically,

$$\mathbf{s}_t = f(y_{t-1}, \mathbf{s}_{t-1}, \mathbf{c}) \tag{2}$$

$$y_t \sim \mathbf{p}_t = \mathsf{softmax}(\mathbf{s}_t, \mathbf{c}) \tag{3}$$

where $\mathbf{s}_t$ is the hidden state of the decoder at time t, y_{t-1} is the previously generated word and $\mathbf{p}_t$ is the output probability distribution at time t.
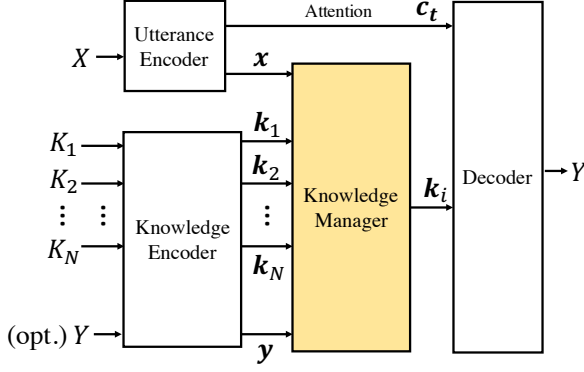
Figure 1: Architecture Overview

Attention. Recently, the attention mechanism (Bahdanau et al., 2014) becomes very popular due to its effectiveness to improve the generation quality. Specifically, in Seq2Seq with attention, each y_t is generated according to a particular context vector $\mathbf{c}_t$ (instead of the same $\mathbf{c}$ used in Equation (2)) where $\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \mathbf{h}_i$. Intuitively, $\mathbf{c}_i$ can be regarded as a weighted sum of the hidden states of the encoder where the weight $\alpha_{t,i}$ measures the relevancy between state $\mathbf{s}_{t-1}$ and $\mathbf{h}_i$.

2.2 Architecture Overview

The architecture overview of our conversation model is presented in Figure 1 and it consists of the following four major components:

- **Utterance Encoder.** The utterance encoder encodes the source utterance X into a vector $\mathbf{x}$, and feeds it into the knowledge manager.

- **Knowledge Encoder.** The knowledge encoder takes as input each knowledge K_i and encodes it into a knowledge vector $\mathbf{k}_i$. When the target utterance Y is available (in training), it also encodes Y into $\mathbf{y}$, which is used later to guide knowledge selection.

- **Knowledge Manager.** The knowledge manager is the most important component in our model. Given the previously generated $\mathbf{x}$ and $\{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_N\}$ (and $\mathbf{y}$ if available), the knowledge manager is responsible to sample a proper knowledge $\mathbf{k}_i$ and feeds it (together with the context vector $\mathbf{c}_t$ attended on the hidden states of the encoder) into the decoder.

- **Decoder.** Finally, the decoder generates responses based on the selected knowledge $\mathbf{k}_i$ and the attention-based context $\mathbf{c}_t$.

In the following, we present the encoders (both utterance and knowledge) in Section 2.3. The knowledge manager is discussed in Section 2.4 while the decoder is described in Section 2.5. Finally, the loss function is elaborated in Section 2.6.

2.3 Encoder

In our encoders, we implement the non-linear transformation f in Equation (1) using a bidirectional RNN with a gated recurrent unit (GRU) (Cho et al., 2014a), which consists of two parts: a forward RNN and a backward RNN.

In the utterance encoder, given the source utterance $X = x_1 \ldots x_n$, the forward RNN reads X from left to right and then, obtains a left-to-right hidden state $\overrightarrow{\mathbf{h}}_t$ for each x_t while the backward RNN reads X in a reverse order, and similarly, obtains a right-to-left hidden state $\overleftarrow{\mathbf{h}}_t$ for each x_t. These two hidden states are concatenated to form an overall hidden state $\mathbf{h}_t$ for x_t. Mathematically,

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t] = [\mathsf{GRU}(x_t, \overrightarrow{\mathbf{h}}_{t-1}); \mathsf{GRU}(x_t, \overleftarrow{\mathbf{h}}_{t+1})]$$

where $[\cdot; \cdot]$ represents a vector concatenation. To obtain a representation $\mathbf{x}$ of the source utterance X, we utilize the hidden states and define $\mathbf{x} = [\overrightarrow{\mathbf{h}}_T; \overleftarrow{\mathbf{h}}_1]$. This vector will be fed into the knowledge manager for knowledge sampling and it will also serve as the initial hidden state of the decoder.

Our knowledge encoder follows the same architecture as the utterance encoder, but they do not share any parameters. Specifically, it encodes each knowledge K_i (and the target utterance Y if it is available) into a vector representation $\mathbf{k}_i$ (and $\mathbf{y}$, respectively) using a bidirectional RNN, and uses it later in the knowledge manager.

2.4 Knowledge Manager

Given the encoded source $\mathbf{x}$ and the encoded knowledge $\{\mathbf{k}_1, \ldots, \mathbf{k}_N\}$, the goal of the knowledge manager is to sample a proper $\mathbf{k}_i$ from the knowledge collection. When the target utterance $\mathbf{y}$ is available, it is also utilized to teach the model for obtaining $\mathbf{k}_i$. The detailed architecture of knowledge manager is illustrated in Figure 2.

Consider the training phase where both X and Y are fed into our model. We compute a conditional probability distribution $\mathbf{p}(\mathbf{k}|\mathbf{x}, \mathbf{y})$ over the knowledge collection and use this distribution to sample $\mathbf{k}_i$. Specifically, we define

$$p(\mathbf{k} = \mathbf{k}_i|\mathbf{x}, \mathbf{y}) = \frac{\exp(\mathbf{k}_i \cdot \mathsf{MLP}([\mathbf{x}; \mathbf{y}]))}{\sum_{j=1}^{N} \exp(\mathbf{k}_j \cdot \mathsf{MLP}([\mathbf{x}; \mathbf{y}]))}$$
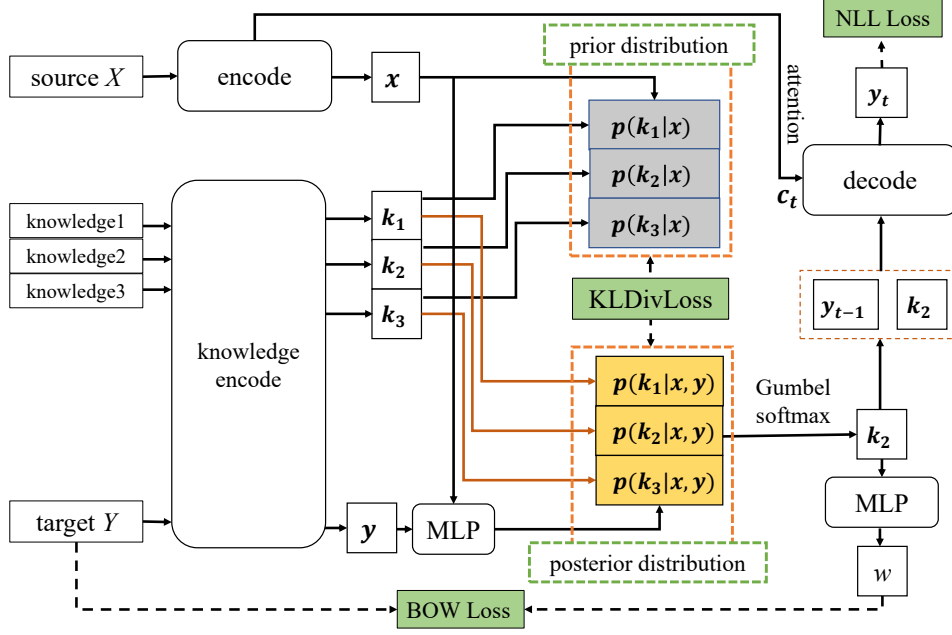
Figure 2: Knowledge Manager and Loss Functions

where $\text{MLP}(\cdot)$ is a fully connected layer. Intuitively, we use the dot product to measure the association between $\mathbf{k}_i$ and the input utterances. A higher association means that $\mathbf{k}_i$ is more relevant and $\mathbf{k}_i$ is more likely to be sampled. Since $\mathbf{p}(\mathbf{k}|\mathbf{x}, \mathbf{y})$ is conditioned on both $\mathbf{x}$ and $\mathbf{y}$, it is a *posterior knowledge distribution* since the actual knowledge used in the target utterance Y can be captured via this distribution.

To sample knowledge according to $\mathbf{p}(\mathbf{k}|\mathbf{x}, \mathbf{y})$, we use Gumbel-Softmax re-parametrization (Jang et al., 2016) (instead of the exact sampling) since it allows backpropagation in non-differentiable categorical distributions. The sampled knowledge will be very helpful in generating the desired response since it contains the knowledge information used in the actual response.

Unfortunately, when considering the evaluation phase or when generating responses, the target utterance Y is not available and thus, we cannot use the posterior distribution to sample knowledge. Alternatively, a *prior knowledge distribution*, denoted by $\mathbf{p}(\mathbf{k}|\mathbf{x})$, is used to estimate the desired $\mathbf{p}(\mathbf{k}|\mathbf{x}, \mathbf{y})$. Specifically,

$$p(\mathbf{k} = \mathbf{k}_i|\mathbf{x}) = \frac{\exp(\mathbf{k}_i \cdot \mathbf{x})}{\sum_{j=1}^{N} \exp(\mathbf{k}_j \cdot \mathbf{x})}.$$

When evaluating or generating responses, $\mathbf{k}_i$ is sampled based on the prior distribution $\mathbf{p}(\mathbf{k}|\mathbf{x})$.

Clearly, we want the prior distribution $\mathbf{p}(\mathbf{k}|\mathbf{x})$

to be as close to the posterior distribution $\mathbf{p}(\mathbf{k}|\mathbf{x}, \mathbf{y})$ as possible so that our model is capable to capture the correct knowledge even without the target utterance Y. For this purpose, we introduce an auxiliary loss, namely the Kullback-Leibler divergence loss (KLDivLoss), to measure the proximity between $\mathbf{p}(\mathbf{k}|\mathbf{x})$ and $\mathbf{p}(\mathbf{k}|\mathbf{x}, \mathbf{y})$. The KLDivLoss is formally defined in the following.

KLDivLoss. We define the KLDivLoss to be

$$\mathcal{L}_{KL}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} p(\mathbf{k} = \mathbf{k}_i|\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{k} = \mathbf{k}_i|\mathbf{x}, \mathbf{y})}{p(\mathbf{k} = \mathbf{k}_i|\mathbf{x})}$$

where θ denotes the model parameters.

Intuitively, when minimizing KLDivLoss, the posterior distribution $\mathbf{p}(\mathbf{k}|\mathbf{x}, \mathbf{y})$ can be regarded as labels and our model is instructed to use the prior distribution $\mathbf{p}(\mathbf{k}|\mathbf{x})$ to approximate the posterior distribution $\mathbf{p}(\mathbf{k}|\mathbf{x}, \mathbf{y})$ accurately. As a consequence, even when the the posterior distribution is unknown when generating responses (since the actual target utterance Y is unknown in real cases), the prior distribution $\mathbf{p}(\mathbf{k}|\mathbf{x})$ can be effectively utilized to sample correct knowledge so as to generate proper responses. To the best of our knowledge, we are the first neural model, which incorporates the posterior knowledge distribution as a guidance, enabling accurate knowledge lookups and high quality response generation.

2.5 Decoder

Conditioned on the context c_t and the selected k_i, our decoder generates response word by word sequentially. Different from the traditional Seq2Seq decoder with attention, we incorporates knowledge into response generation. To achieve this, we introduce two variants of decoders. The first one is a "hard" knowledge introducing decoder with a standard GRU and concatenated inputs and the second one is "soft" decoder with a hierarchical gated fusion unit (Yao et al., 2017).

Standard GRU with Concatenated Inputs. Let s_t be the last hidden state, y_{t-1} be the word generated in the last step and c_t be the attention-based context vector on the hidden states $h_1, h_1, \ldots, h_n$ of the encoder. The current hidden state is:

$$s_t = \text{GRU}([y_{t-1}; k_i], s_{t-1}, c_t)$$

where we concatenate y_{t-1} with the selected k_i. This is a simple and intuitive decoder. However, it forces knowledge k_i to attend decoding, which is less flexible and is not desirable in some scenarios.

Hierarchical Gated Fusion Unit (HGFU). HGFU provides a softer way to incorporate knowledge into response generation and it consists of three major components, namely a utterance GRU, a knowledge GRU and a fusion unit.

The former two components follow the standard GRU structure. They produce the hidden representations for the last generated word y_{t-1} and the selected knowledge k_i, respectively. Specifically,

$$s_t^y = \text{GRU}(y_{t-1}, s_{t-1}, c_t)$$

$$s_t^k = \text{GRU}(k_i, s_{t-1}, c_t)$$

Then, the fusion unit combines these two hidden states to generate the hidden state of the decoder at time t (Yao et al., 2017). Mathematically,

$$s_t = r \odot s_t^y + (1 - r) \odot s_t^k$$

where $r = \sigma(W_z[\tanh(W_y s_t^y); \tanh(W_k s_t^k)])$ and W_z, W_y and W_k are parameters. Intuitively, the gate r controls the amount of contributions from s_t^y and s_t^k to the final hidden state s_t, allowing a flexible knowledge incorporation schema.

After obtaining the hidden state s_t, the next word y_t is generated according to Equation (3).

2.6 Loss Function

Apart from the KLDivLoss introduced in Section 2.4, two additional loss functions are used and they are the NLL loss and the BOW loss. The NLL loss captures the *ordering* information while the BOW loss captures the *bag-of-word* information. All loss functions are also elaborated in Figure 2.

NLL Loss. The objective of NLL loss is to quantify the difference between the actual response and the response generated by our model. It minimize the Negative Log-Likelihood (NLL) :

$$\mathcal{L}_{NLL}(\theta) = -\frac{1}{m} \sum_{t=1}^{m} \log p_\theta(y_t | y_{<t}, x, k_i)$$

where θ denotes the model parameters and $y_{<t}$ denotes the previously generated words.

BOW Loss. The BOW loss (Zhao et al., 2017) is designed to ensure the accuracy of the sampled knowledge k_i by enforcing the relevancy between the knowledge and the target response. Specifically, let $w = \text{MLP}(k_i) \in \mathcal{R}^{|V|}$ where $|V|$ is the vocabulary size, and we define

$$p_\theta(y_t | k_i) = \frac{\exp(w_{y_t})}{\sum_{v \in V} \exp(w_v)}.$$

Then, the BOW loss is defined to minimize

$$\mathcal{L}_{BOW}(\theta) = -\frac{1}{m} \sum_{t=1}^{m} \log p_\theta(y_t | k_i).$$

In summary, unless specified explicitly, the total loss of a given a training example $(X, Y, \{K_i\}_{i=1}^N)$ is the sum of the KLDivLoss, the NLL loss and the BOW loss. That is,

$$\mathcal{L}(\theta) = \mathcal{L}_{KL}(\theta) + \mathcal{L}_{NLL}(\theta) + \mathcal{L}_{BOW}(\theta).$$

3 Experiments

3.1 Dataset

We conducted experiments on two recently created datasets in the literature, namely the Persona-chat dataset (Zhang et al., 2018) and the Wizard-of-Wikipedia dataset (Dinan et al., 2018).

Persona-chat. In the Persona-chat dataset, each dialogue was constructed from a randomly pair of crowd-workers, who were instructed to chat to know each other. To produce meaningful and interesting conversations, each worker was assigned a persona profile, describing their characteristic, and this persona profile serves as the knowledge

in the conversation (see Table 1 for an example). There are 151,157 turns (each turn corresponds to a source and a target utterances pair) of conversations in the Persona-chat dataset, which we divide into 122,499 for train, 14,602 for validation and 14,056 for test. The average size of a knowledge collection (i.e., the average number of sentences in a persona profile) in this dataset is 4.49.

Wizard-of-Wikipedia. Wizard-of-Wikipedia is a chit-chatting dataset between two agents on some chosen topics. One of the agent, which is also known as the wizard, plays the role of a knowledge expert and he has the access to an information retrieval system for acquiring knowledge. The other agent acts as a curious learner. From this dataset, 76,236 turns of conversations are obtained and 68,931/3,686/3,619 of them are used for train/validation/test. The average size of a knowledge collection accessed by a wizard is 67.57. The large amount of candidate knowledge in this dataset implies that identifying in the correct knowledge is difficult and challenging.

3.2 Models for Comparison

We implemented our model, namely the _Posterior Knowledge Selection (PostKS)_ model, for evaluation. In particular, two variants of our models were implemented to demonstrate the effect of different ways of incorporating knowledge:

- **PostKS(concat)** is the "hard" knowledge introducing model with a GRU decoder where knowledge is concatenated as input.

- **PostKS(fusion)** is the "soft" knowledge introducing model where knowledge is incorporated with a hierachical gated fusion unit.

In addition, we compared our models with the following three state-of-the-art baselines:

- **Seq2Seq**: a Seq2Seq model with attention that does not have knowledge information (Shang et al., 2015; Vinyals and Le, 2015).

- **MemNet(hard)**: a Bag-of-Words memory network adapted from (Ghazvininejad et al., 2018), where knowledge is sampled from the collection and fed into the decoder.

- **MemNet(soft)**: a soft knowledge-grounded model adapted from (Ghazvininejad et al., 2018). The memory units store knowledge, which are attended in the decoder.

Among them, Seq2Seq is compared for demonstrating the effect of introducing knowledge in conversation generation while the MemNet based models are compared to verify that our models are more capable of incorporating correct knowledge than the existing knowledge-based models.

3.3 Implementation Details

The encoders and decoders in our model have 2-layer GRU structures with 800 hidden states for each layer, but they do not share any parameters. We set the word embedding size to be 300 and initialize it using GloVe (Pennington et al., 2014). The vocabulary size is set to be 20,000. We used the the Adam optimizer with a mini-batch size of 128 and we set the learning rate to be 0.0005.

We trained our model with at most 20 epochs on a P40 machine. In the first 5 epochs, we minimize the BOW loss only for pre-training the knowledge manager. In the remaining epochs, we minimize over the sum of all losses (KLDiv, NLL and BOW). After each epoch, we save a model and the model with the minimum loss is selected for evaluation. Upon acceptance, our models and datasets will be available online[1].

3.4 Automatic and Human Evaluation

We adopted several automatic metrics to evaluate the performance of each model and the result is summarized in Table 2. Among them, _BLEU-1/2/3_ and _Distinct-1/2_ are two popular metrics for evaluating the quality and diversity of the generated responses. _Knowledge R/P/F1_ is a metric adapted from (Dinan et al., 2018), which measures the unigram recall/precision/F1 score between the generated responses and the knowledge collection. Specifically, given the set of non-stopwords in Y and the knowledge collection $\{K_i\}_{i=1}^{N}$, denoted by W_Y and W_K, respectively, we define Knowledge R(ecall) and Knowledge P(recision) to be

$$\frac{|W_Y \cap W_K|}{|W_K|} \text{ and } \frac{|W_Y \cap W_K|}{|W_Y|}$$

and Knowledge F1 $= 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$.

As shown in Table 2, our models outperform all baselines _significantly_ by achieving the _highest_ scores in _all_ automatic metrics. Specifically, compared with traditional Seq2Seq, incorporating knowledge is shown to be very helpful in generating meaningful and diverse responses.

---
[1]https://github.com/ifr2/PostKS

Dataset	Model	Automatic Evaluation			Human Evaluation
		BLEU-1/2/3	Distinct-1/2	Knowledge R/P/F1	
Persona-chat	Seq2Seq	0.182/0.093/0.055	0.026/0.074	0.0042/0.0172/0.0066	0.66
	MemNet(hard)	0.186/0.097/0.058	0.037/0.099	0.0115/0.0430/0.0175	0.74
	MemNet(soft)	0.177/0.091/0.055	0.035/0.096	0.0146/0.0567/0.0223	0.76
	PostKS(concat)	0.182/0.096/0.057	**0.048**/0.126	0.0365/0.1486/0.0567	0.89
	PostKS(fusion)	**0.190/0.098/0.059**	0.046/**0.134**	**0.0574/0.2137/0.0870**	**0.93**
Wizard-of-Wikipedia	Seq2Seq	0.169/0.066/0.032	0.036/0.112	0.0069/0.5780/0.0136	0.83
	MemNet(hard)	0.159/0.062/0.029	0.043/0.138	0.0077/0.6036/0.0151	0.89
	MemNet(soft)	0.168/0.067/**0.034**	0.037/0.115	0.0076/0.6713/0.0151	0.92
	PostKS(concat)	0.167/0.066/0.032	**0.056**/0.209	0.0080/0.6979/0.0158	0.95
	PostKS(fusion)	**0.172/0.069/0.034**	**0.056/0.213**	**0.0088/0.7047/0.0174**	**0.97**

Table 2: Automatic and Manual Evaluation on Persona-chat and Wizard-of-Wikipedia

For example, the Distinct-1/2 metric on Persona-chat is increased from 0.026/0.074 (Seq2Seq) to 0.048/0.126 (PostKS(concat)), meaning that the diversity of the responses is greatly improved by augmenting with knowledge. Besides, when comparing with baselines which also leverage knowledge, our models demonstrate their ability on properly incorporating the correct knowledge in response generation. In particular, comparing our PostKS(fusion) model against the MemNet(soft) model (both of them are soft knowledge-grounded models), we achieve higher BLEU scores and higher Distinct scores. This is because that the posterior knowledge is fully utilized in our models to provide an effective guidance on obtaining the correct knowledge, resulting in responses with higher quality and higher diversity. Note that, compared with knowledge selection on Persona-chat, the task of locating knowledge correctly on Wizard-of-Wikipedia is more challenging due to a larger knowledge collection size. Nevertheless, the performance of our models is still consistently better than all baselines in this case. For example, PostKS(fusion) has higher knowledge R/P/F1 compared with all MemNet based models on Wizard-of-Wikipedia, indicating that it is able to not only obtain the correct knowledge, but also ensure that knowledge is truly incorporated in the response generated. Finally, we observe that PostKS(fusion) performs slightly better than PostKS(concat) in most cases. This verifies that soft knowledge incorporation is a better way of introducing knowledge to conversation generation since it allows for more flexible knowledge integration and less sensitivity to noisy information.

We also performed point-wise human evalua-
tion on our models. Specifically, three annotators were recruited and they were asked to rate the overall quality of the responses generated by each model. The rating ranges from 0 to 2, where 0 means that the response is completely irrelevant, 1 means that the response is acceptable but not very informative, and 2 means that the response is natural, relevant and informative. We randomly sampled 200 responses for each model on each dataset, resulting in 2,000 responses in total for human annotation. We reported the average rating from all annotators and the result is shown in the last column in Table 2. The agreement ratio, computed with Fleiss' kappa (Fleiss, 1971), is 0.48 and 0.41 on Persona-chat and Wizadr-of-Wikipedia, respectively, showing moderate agreement. According to the result, both of our models, PostKS(concat) and PostKS(fusion), are remarkably better than all baselines in terms of human rating, demonstrating the effectiveness of our posterior knowledge selection techniques.

3.5 Case Study

Table 3 shows two typical examples of the generated responses on both Persona-chat and Wizard-of-Wikipedia. For the lack of space, we only display four pieces of knowledge from each dataset for illustration. In the example from Persona-chat, the source utterance is asking whether the agent likes *music* or not. Without the access to knowledge, Seq2Seq produces a generic response which is irrelevant and does not contain any useful information. MemNet(hard) tries to incorporate knowledge in the generated response, but, unfortunately, it samples the wrong knowledge, leading to an irrelevant response about *broc-*

	Persona-chat	Wizard-of-Wikipedia
Source	I like all music how about you?	Cool! You sure know your stuff about country music!
Knowledge	K1. I hate broccoli. K2. Rock music is my favorite. K3. I am afraid of the dark. K4. My mom is my best friend.	K1. George Glenn Jones (September 12, 1931 April 26, 2013) was an American musician, singer and songwriter. K2. In 2009, in the United States country music was the most listened to rush hour radio genre. K3. Country (or country and western) is a musical genre that originated in the southern United States in the early 1920s. K4. Bro-country is a subgenre of mainstream country music originating in the second decade of the 21st century that is influenced by 21st-century hip hop, hard rock and electronic music.
Seq2Seq	I am good, how are you?	Yes, i know country music.
MemNet (hard)	I don't like broccoli, what about you?	I love rock music. It was one of the most popular rock bands in the uk.
MemNet (soft)	I like all kinds of music. What do you do?	I do know that country music originated in the united states.
PostKS (concat)	I like to listen to rock music.	I love george glenn. He was an american singer and songwriter.
PostKS (fusion)	I love rock music. What is your favorite band?	I do like country music. It is the most listened to rush hour radio genre.

Table 3: Examples of the Generated Responses on Persona-chat and Wizard-of-Wikipedia

coli rather than *music*. In comparison, the remaining three models generate responses with the help of the correct knowledge. Among them, our PostKS(fusion) and PostKS(concat) models perform better since they are more specific and informative by mentioning exactly the *rock music*. In particular, our soft knowledge introducing model, PostKS(fusion), performs noticeably well since it does not only answer questions, but also raises a relevant question about the *favorite band*, allowing evolving and interactive conversations. The example from Wizard-of-Wikipedia is a conversation about *country music*. Similar to Persona-chat, our models enjoy superior performance by producing more informative and relevant responses.

4 Related Work

The great success of Seq2Seq model motivates the development of various techniques for improving the quality and diversity of generated responses. Examples include diversity promotion (Li et al., 2016) and unknown words handling (Gu et al., 2016). However, the problem of tending to generate short and generic words still remains in these models since they do not have the access to any external information.

Recently, knowledge incorporation is shown to be an effective way to improve the performance of conversation models. Both unstructured text and structured triples can be utilized as knowledge. Long et al. (2017) obtains knowledge from unstructured texts using a convolutional neural network. Ghazvininejad et al. (2018) stores texts as knowledge in a memory network and uses them to produce more informative responses. A neural knowledge diffusion model was also proposed in (Liu et al., 2018), where the model is augmented with convergent and divergent thinking over a knowledge base. In addition, large scale commonsense knowledge bases were first utilized in (Zhou et al., 2018) for conversation generation and many domain-specific knowledge bases were also considered to ground neural models with knowledge (Xu et al., 2017; Zhu et al., 2017; Gu et al., 2016).

However, none of the existing models have demonstrated their ability on (1) locating the correct knowledge in a large candidate knowledge set and (2) properly utilizing the knowledge in generating high quality responses. Instead, they condition knowledge simply on the conversation history. In comparison, our model is *different* from all existing ones since it makes full use of the posterior knowledge distribution and thus, our model is effectively taught to select the appropriate knowledge and to ensure that the knowledge is truly utilized in generating responses.

5 Conclusion

In this paper, we present a novel open-domain conversation model, which is the first neural model that makes use of posterior knowledge distribution to facilitate knowledge selection and incorporation. In particular, even when the posterior information is not available, our model can still pro-

duce appropriate and informative responses. Extensive experiments on both automatic and human metrics demonstrate the effectiveness and usefulness of our model. As for future work, we plan to take the advantage of reinforcement learning into our model for even better performance.

References

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

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

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

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *ACL*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. *NAACL*.

Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge

diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1489–1498.

Yinong Long, Jianan Wang, Zhen Xu, Zongsheng Wang, Baoxun Wang, and Zhuoran Wang. 2017. A knowledge enhanced generative conversational service agent. In *Proceedings of the 6th Dialog System Technology Challenges (DSTC6) Workshop*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2017. Incorporating loose-structured knowledge into conversation modeling via recall-gate lstm. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 3506–3513. IEEE.

Lili Yao, Yaoyuan Zhang, Yansong Feng, Dongyan Zhao, and Rui Yan. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2190–2199.

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 preprint arXiv:1801.07243*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.

Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*.