DukeNet: A Dual Knowledge Interaction Network for Knowledge-Grounded Conversation

Anonymous Author(s)
Submission Id: 74

ABSTRACT

Today's conversational agents often generate responses that not sufficiently informative. One way of making them more informative is through the use of of external knowledge sources with so-called Knowledge-Grounded Conversations (KGCs). In this paper, we target the *Knowledge Selection* (KS) task, a key ingredient in KGC, that is aimed at selecting the appropriate knowledge to be used in the next response. Existing approaches to Knowledge Selection (KS) based on learned representations of the conversation context, that is previous conversation turns, and use Maximum Likelihood Estimation (MLE) to optimize KS. Such approaches have two main limitations. First, they do not explicitly track what knowledge has been used in the conversation nor how topics have shifted during the conversation. Second, MLE often relies on a limited set of example conversations for training, from which it is hard to infer that facts retrieved from the knowledge source can be re-used in multiple conversation contexts, and vice versa.

We propose Dual Knowledge Interaction Network (DukeNet), a framework to address these challenges. DukeNet explicitly models *knowledge tracking* and *knowledge shifting* as dual tasks. We also design Dual Knowledge Interaction Learning (DukeL), an unsupervised learning scheme to train DukeNet by facilitating interactions between knowledge tracking and knowledge shifting, which, in turn, enables DukeNet to explore extra knowledge besides the knowledge encountered in the training set. This dual process also allows us to define rewards that help us to optimize both knowledge tracking and knowledge shifting. Experimental results on two public KGC benchmarks show that DukeNet significantly outperforms state-of-the-art methods in terms of both automatic and human evaluations, indicating that DukeNet enhanced by DukeL can select more appropriate knowledge and hence generate more informative and engaging responses.

ACM Reference Format:

Anonymous Author(s). 2020. DukeNet: A Dual Knowledge Interaction Network for Knowledge-Grounded Conversation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Xi'an, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Open-domain conversational agents (a.k.a. chatbots) aim to satisfy human needs for information, communication, entertainment and more [1, 11, 47]. The development of such agents has benefited significantly from advances in sequence-to-sequence learning [30, 34]. However, a serious problem of vanilla sequence-to-sequence based conversational models is that they tend to generate dull and non-informative responses [7, 39], such as "*I don't know*" and "*thank you*." To address this problem of generating uninformative responses, the Knowledge-Grounded Conversation (KGC) task has been introduced; it leverages external knowledge to enhance open-domain conversational models [6, 20, 28]. As shown in Fig. 1, given a conversation context and external knowledge pool (with a large set of knowledge sentences), the goal of KGC is to generate informative and engaging responses by referring to relevant knowledge. KGC can be divided into two sequential subtasks: (1) *Knowledge Selection* (KS): to select the appropriate knowledge at the current turn from a knowledge pool; and (2) Response Generation (RG): to generate a natural language response based on the selected knowledge and conversation context. Of the two subtasks, Knowledge Selection (KS) is of vital importance, as it decides what to be talked in the response, and selecting inappropriate knowledge will directly result in an inappropriate response [15].

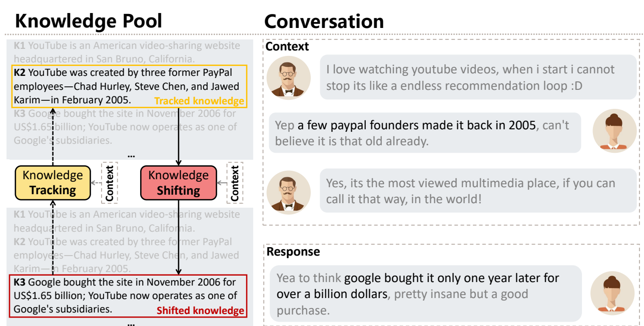

Figure 1: An example of a knowledge-grounded conversation from the Wizard of Wikipedia dataset [6].

There is a growing number of studies on KS and promising results have been achieved [see, e.g., 6, 8]. In terms of modeling, most approaches try to predict the next knowledge based on representations of the conversation context [15]. In terms of learning, most approaches optimize their models via Maximum Likelihood Estimation (MLE) based on conversations encountered in the training set [12]. Previous work fails to address two important characteristics of KS for conversations.

(1) *knowledge tracking* (ground the knowledge that has been talked about to the conversation context) and *knowledge shifting* (select the knowledge to be talked about next) are not explicitly modeled. Explicitly modeling knowledge tracking and shifting allows us to better capture the interaction between the knowledge at adjacent turns. Consider, for example, in Fig. 1, for the

SIGIR '20, July 25–30, 2020, Xi'an, China

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

knowledge K2 "YouTube was created by three former PayPal employees…" and the knowledge K3 "Google bought the site in November 2006 for US$1.65 billion…." If we know that K2 has been used in the conversation context, it is natural to use K3 in the next response.

(2) Unlike Question Answering (QA) tasks in which each query has only one unique answer in most cases [26, 27], Kim et al. [12] show that one-to-many mapping between context and knowledge is common in KGC, i.e., for a given conversation context, we can choose different knowledge sentences to form different responses. In fact, there is also many-to-one mapping between context and knowledge, i.e. the same knowledge sentence can be used in different conversation contexts. Thus, this forms a many-to-many mapping. For instance, given the context in Fig. 1, it is also appropriate to select other knowledge besides K3, e.g., the most popular video or the most famous video creator. Conversely, it is also appropriate to use K3 in Fig. 1 in other contexts such as "I wonder what the relationship between Google and Youtube is" or "Do you know the history of Youtube." Unfortunately, for a given conversation context, only one knowledge sentence is encountered in existing datasets [6, 21], because it is really hard to collect all possible knowledge sentences for a certain context when constructing a KGC dataset. Existing studies into KGC rely on MLE to train their models, while MLE only considers the demonstrated knowledge as ground truth, which is restrictive.

To address the issues listed above, we propose a novel framework, Dual Knowledge Interaction Network (DukeNet), that explicitly models *knowledge tracking* and *knowledge shifting* in conversations. Besides, we further formulate knowledge tracking and knowledge shifting as dual tasks in order to better facilitate interaction between them. Dual Knowledge Interaction Learning (DukeL) enables knowledge tracking and shifting to teach each other in an unsupervised learning way without external supervision. We alternate these two dual processes until convergence. During the two dual processes, DukeL will explore and reward extra appropriate knowledge that is not manifest in the training set, but which help address the many-to-many mapping phenomenon in conversations.

However, there are incompatible dual processes between training and inference. Specifically, during inference we can only execute knowledge tracking and shifting in order, thus knowledge tracking cannot get the shifted knowledge as input. To alleviate this problem, we further distinguish knowledge tracking as *prior and posterior knowledge tracking*, where the former only takes context as input, while the latter additional takes the shifted knowledge as input. During training, besides optimizing the posterior knowledge tracking and knowledge shifting based on the closed loop between them, we force the prior knowledge tracking to approximate the output of posterior knowledge tracking to get benefit. During inference, we only execute prior knowledge tracking and shifting.

Experiments on the Wizard of Wikipedia [6] and Holl-E [21] datasets show that DukeNet can select more appropriate knowledge and hence generate more informative and engaging responses by explicitly modeling knowledge tracking and knowledge shifting, and formulating their interactions as dual learning.

The contributions of this paper are summarized as follows:

- We propose a novel framework, DukeNet, which explicitly models *knowledge tracking* and *knowledge shifting* as dual tasks to promote KS.
- We devise DukeL, which introduces an unsupervised learning scheme for KS.
- We conduct automatic and human evaluations on two benchmark datasets, which shows that DukeNet outperforms recent state-of-the art methods, and can select more appropriate knowledge to generate more informative and engaging responses.

2 RELATED WORK

We survey two types of related work: Knowledge-Grounded Conversations (KGCs) and dual learning.

2.1 Knowledge-grounded conversation

Existing methods on KGC can be divided into two categories: *structured knowledge based* and *unstructured knowledge based*. The former conditions response generation on knowledge triples [17, 37, 49], while the latter conditions on free knowledge text [24]. Unstructured knowledge based approaches to KGC can be further divided into *document based* (where they are given whole documents, e.g., Reddit articles) [14, 21, 25, 50] and *sentence based* (where they are given separate sentences, e.g., Foursquare tips) [6, 8, 12, 15]. In this paper, we focus on sentence based KGC. Next, we briefly introduce recent advances in this direction.

Ghazvininejad et al. [8] regard Foursquare tips as knowledge sentences and propose MemNet (a variant of Memory Network [32]) which stores the latent representations of knowledge sentences in a memory module such that a SeqSeq model can attentively select useful knowledge from it to generate responses. Dinan et al. [6] collect a large sentence based benchmark dataset, Wizard of Wikipedia, which retrieves Wikipedia articles and then flattens all the articles into separate sentences and clearly labels the ground truth knowledge sentence used in each response. They also propose Transformer MemNet (TMemNet) that improves MemNet by replacing RNN with Transformers [38] and introduce a KS loss to supervise the KS process. Lian et al. [15] propose Posterior Knowledge Selection (PostKS), which uses response and context to predict a distribution over knowledge and regards this distribution as posterior knowledge distribution. They use that as pseudo-label to guide KS during training process. Kim et al. [12] propose Sequential Knowledge Transformer (SKT), which sequentially models the history of KS of previous turns via a sequential latent variable model [31] to promote KS at the current turn. SKT gets the leading performance on the Wizard of Wikipedia dataset at the time of writing [6]. Although SKT also models the knowledge used in conversation history, it does not explicitly distinguish knowledge tracking and shifting and still uses MLE to train the model based on the limited demonstrated examples in training set.

2.2 Dual learning

Dual learning has been successfully applied to many tasks, such as machine translation [10, 41], question answering [33, 36], conversation [3, 46], text style transfer [19] as well as image-to-image translation [45]. The core idea of dual learning is to take advantage

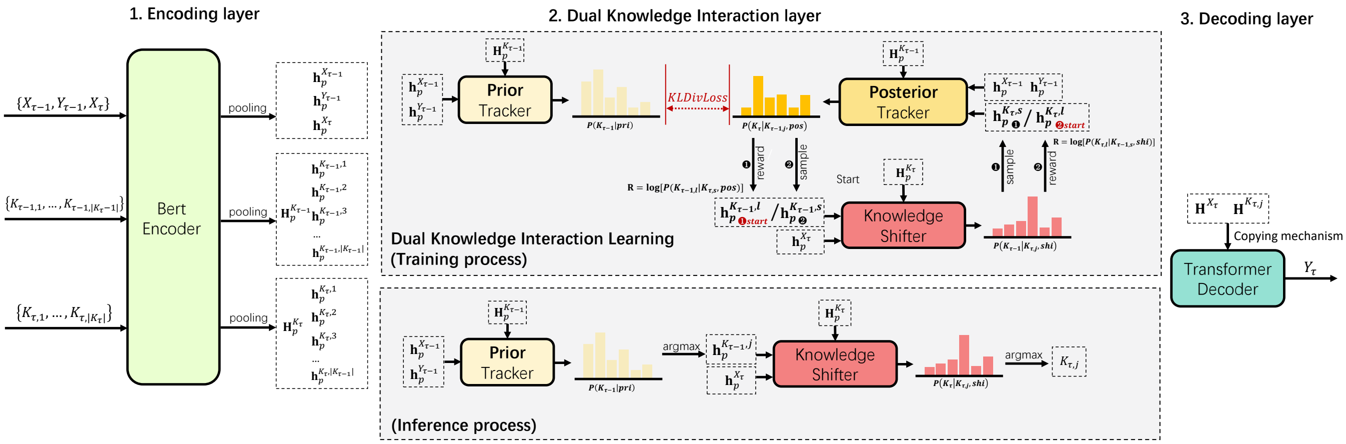

Figure 2: An overview of DukeNet. Section 3 contains a walkthrough of the model.

of the closed loop between the forward agent f (mapping from domain X to domain Y) and backward agent g (mapping from domain Y to X) to improve the performances of each other.

There are many paradigms in dual learning. He et al. [10] first propose dual learning and apply it to machine translation, which executes dual learning by maximizing the likelihood of data reconstruction. Concretely, $x \in X$ is first fed to the forward agent f to output $\hat{y} \in Y$, and $\hat{y}$ is then fed to the backward agent g to output $\hat{x} \in X$. The distortion between x and $\hat{x}$ is used as a reconstruction reward to optimize the forward agent f. Similarly, the reconstruction reward between y and $\hat{y}$ can be used to optimize the backward agent g. Xia et al. [43] propose dual supervised learning, which introduces a duality regularization term in their loss function. The special term reflects the probabilistic correlation between two agents to better guide the training. Xia et al. [42] argue that existing work only considers the duality during the training process and further a dual inference framework, which enables dual agents to improve each other during the inference process without re-training. Xia et al. [44] propose model-level dual learning, which shares the model parameters playing similar roles in the two agents. Wang et al. [40] propose multi-agent dual learning, which introduces more agents in the two directions respectively to maximize the likelihood of data reconstruction. They show that more agents can lead to more reliable and robust reconstruction reward.

Unlike the work listed above, we regard knowledge tracking and shifting in KGC as two dual tasks and propose DukeL to let the two tasks teach each other by maximizing the likelihood of knowledge reconstruction. Especially, we present prior and posterior knowledge tracking in DukeL to solve the incompatible dual processes issue during training and inference.

3 METHOD

3.1 KGC formulation

We use $D = \{(X_\tau, Y_\tau)\}_{\tau=1}^{|D|}$ to denote the set of conversation turns, where (X_τ, Y_τ) is a conversation turn from two distinct speakers. At turn τ, given the conversation context $C_\tau = (X_{\tau-1}, Y_{\tau-1}, X_\tau)$ (we use a single previous turn and the current turn from the first speaker X_τ as conversation context), and corresponding knowledge pool $K_\tau = \{K_{\tau,j}\}_{j=1}^{|K_\tau|}$ with $|K_\tau|$ knowledge sentences (which are retrieved w.r.t. the conversation context), the task of KGC is to generate a response $Y_\tau = (y_{\tau,1}, y_{\tau,2}, \ldots, y_{\tau,|Y_\tau|})$ with $|Y_\tau|$ tokens. With sequence-to-sequence modeling, this can be formulated as follows:

$$P(Y_\tau \mid C_\tau, K_\tau) = \prod_{t=1}^{|Y_\tau|} P(y_{\tau,t} \mid y_{\tau,<t}, C_\tau, K_\tau), \tag{1}$$

where $y_{\tau,t}$ is the t-th token; $y_{\tau,<t}$ are the tokens up to the $(t-1)$-th decoding step.

3.2 Overview of DukeNet

As shown in Fig. 2, DukeNet consists of three layers: (1) an *encoding layer*, (2) a *dual knowledge interaction layer*, and (3) a *decoding layer*.

The encoding layer employs BERT to encode context C_τ and knowledge pool K_τ into context representations $\mathbf{H}_p^{C_\tau} = (\mathbf{h}_p^{X_{\tau-1}}, \mathbf{h}_p^{Y_{\tau-1}}, \mathbf{h}_p^{X_\tau})$ and knowledge representations $\mathbf{H}_p^{K_\tau} = \{\mathbf{h}_p^{K_{\tau,j}}\}_{j=1}^{|K_\tau|}$, respectively.

The dual knowledge interaction layer contains a *Prior Knowledge Tracker* (Pri), a *Knowledge Shifter* (Shi), and a *Posterior Knowledge Tracker* (Pos). The prior knowledge tracker takes context $\mathbf{h}_p^{X_{\tau-1}}$, $\mathbf{h}_p^{Y_{\tau-1}}$ as inputs to predict a prior knowledge tracking distribution $P(K_{\tau-1}|pri)$ over knowledge pool $K_{\tau-1}$, based on which we can get the tracked knowledge representation $\mathbf{h}_p^{K_{\tau-1,j}}$. The knowledge shifter takes context $\mathbf{h}_p^{X_\tau}$ and tracked knowledge $\mathbf{h}_p^{K_{\tau-1,j}}$ as inputs to predict knowledge shifting distribution $P(K_\tau|K_{\tau-1,j}, shi)$ over knowledge pool K_τ, based on which we can get the shifted knowledge representation $\mathbf{h}_p^{K_{\tau,j}}$. The posterior knowledge tracker takes the shifted knowledge $\mathbf{h}_p^{K_{\tau,j}}$ as input to predict posterior knowledge tracking distribution $P(K_{\tau-1}|K_{\tau,j}, pos)$ over knowledge pool $K_{\tau-1}$. Note that the posterior knowledge tracker is only used during training.

The *decoding layer* contains a Transformer decoder to generate response Y_τ token by token based on the context representation and the shifted knowledge representation.

During training, we devise an unsupervised learning scheme, DukeL, that regards the posterior knowledge tracker and knowledge shifter as dual tasks. Specifically, the shifted knowledge $K_{\tau,s}$

(s denotes random sampling) that is sampled from the knowledge shifting distribution $P(K_\tau|shi)$ can be fed back into the posterior knowledge tracker to recover the tracked knowledge (the original input of knowledge shifter). We regard the recovering probability as a reward to optimize the knowledge shifter. The posterior knowledge tracker can be optimized in a similar manner. The above process forms a closed loop to alternatively train the posterior knowledge tracker and knowledge shifter. Meanwhile, we force the prior knowledge distribution $P(K_{\tau-1}|pri)$ (from the prior knowledge tracker) to get close to the posterior knowledge distribution $P(K_{\tau-1}|K_{\tau,j}, pos)$ (from the posterior knowledge tracker) via Kullback-Leibler Divergence Loss (KLDivLoss) such that the prior knowledge tracker can benefit from the above dual learning process even if it is not involved in the closed loop.

During inference, we only execute the prior knowledge tracker and knowledge shifter to do knowledge tracking and shifting, respectively. Hence the issue of incompatible dual processes that arises during training and inference can be solved effectively. And based on the shifted knowledge, we generate the next response. Next, we introduce the three layers and the learning scheme.

3.3 Encoding layer

We encode the conversation context $C_\tau = (X_{\tau-1}, Y_{\tau-1}, X_\tau)$ into hidden representations $\mathbf{H}_p^{C_\tau} = (\mathbf{h}_p^{X_{\tau-1}}, \mathbf{h}_p^{Y_{\tau-1}}, \mathbf{h}_p^{X_\tau})$ using BERT$_{\text{base}}$ and an average pooling operation [2]:

$$\begin{aligned}
\mathbf{H}^{X_{\tau-1}} &= \text{BERT}(X_{\tau-1}) \in \mathbb{R}^{|X_{\tau-1}|\times d},\ \mathbf{h}_p^{X_{\tau-1}} = \text{p}(\mathbf{H}^{X_{\tau-1}}) \in \mathbb{R}^d,\\
\mathbf{H}^{Y_{\tau-1}} &= \text{BERT}(Y_{\tau-1}) \in \mathbb{R}^{|Y_{\tau-1}|\times d},\ \mathbf{h}_p^{Y_{\tau-1}} = \text{p}(\mathbf{H}^{Y_{\tau-1}}) \in \mathbb{R}^d,\\
\mathbf{H}^{X_\tau} &= \text{BERT}(X_\tau) \in \mathbb{R}^{|X_\tau|\times d},\ \mathbf{h}_p^{X_\tau} = \text{p}(\mathbf{H}^{X_\tau}) \in \mathbb{R}^d,
\end{aligned} \tag{2}$$

where d stands for hidden size and p refers to pooling operation. Similarly, we encode the knowledge sentences in knowledge pool $K_\tau = \{K_{\tau,j}\}_{j=1}^{|K_\tau|}$ into representations $\mathbf{H}_p^{K_\tau} = \{\mathbf{h}_p^{K_{\tau,j}}\}_{j=1}^{|K_\tau|} \in \mathbb{R}^{|K_\tau|\times d}$.

3.4 Dual knowledge interaction layer

3.4.1 Prior knowledge tracker. Given context representations $\mathbf{h}_p^{X_{\tau-1}}$ and $\mathbf{h}_p^{Y_{\tau-1}}$, the prior knowledge tracker predicts the prior knowledge tracking distribution $P(K_{\tau-1}|pri)$ over the knowledge pool $K_{\tau-1}$, which is estimated as follows:

$$\begin{aligned}
P(K_{\tau-1}|pri) &= \text{softmax}(\mathbf{Q}_{pri}\mathbf{K}_{pri}^\top) \in \mathbb{R}^{|K_{\tau-1}|}\\
\mathbf{Q}_{pri} &= \text{mlp}([\mathbf{h}_p^{X_{\tau-1}}; \mathbf{h}_p^{Y_{\tau-1}}]) \in \mathbb{R}^{1\times d}\\
\mathbf{K}_{pri} &= \text{mlp}(\mathbf{H}_p^{K_{\tau-1}}) \in \mathbb{R}^{|K_{\tau-1}|\times d},
\end{aligned} \tag{3}$$

where $\text{mlp}(\cdot) = \cdot\mathbf{W} + \mathbf{b}$ is a Multilayer Perceptron (MLP) and ; denotes the vector concatenation operation.

3.4.2 Knowledge shifter. Given context representation $\mathbf{h}_p^{X_\tau}$ and tracked knowledge representation $\mathbf{h}_p^{K_{\tau-1,j}}$, the knowledge shifter predicts the shifting knowledge distribution $P(K_\tau|K_{\tau-1,j}, shi)$ over knowledge pool K_τ, which is estimated as follows:

$$\begin{aligned}
P(K_\tau|K_{\tau-1,j}, shi) &= \text{softmax}(\mathbf{Q}_s\mathbf{K}_s^\top) \in \mathbb{R}^{|K_\tau|}\\
\mathbf{Q}_{shi} &= \text{mlp}([\mathbf{h}_p^{X_\tau}; \mathbf{h}_p^{K_{\tau-1,j}}]) \in \mathbb{R}^{1\times d}\\
\mathbf{K}_{shi} &= \text{mlp}(\mathbf{H}_p^{K_\tau}) \in \mathbb{R}^{|K_\tau|\times d}.
\end{aligned} \tag{4}$$

Algorithm 1 Dual knowledge interaction learning algorithm

Warm-up training (Phase 1) ▷ see Eq. 12
for each iteration i in *dual interaction training* **do** (Phase 2)
  Sample an example from training set;
  ▷ Start to train knowledge shifter
  Feed $\mathbf{h}_p^{K_{\tau-1,l}}$ to knowledge shifter;
  Get knowledge shifting distribution $P(K_\tau \mid K_{\tau-1,l}, shi)$;
  Sample $\mathbf{h}_p^{K_{\tau,s}}$ from $P(K_\tau \mid K_{\tau-1,l}, shi)$;
  Feed $\mathbf{h}_p^{K_{\tau,s}}$ to posterior knowledge tracker to get reward;
  Update $\theta_1 = [\theta_{embedding}, \theta_{encoder}, \theta_{shi}]$ using reward;
  ▷ Start to train posterior knowledge tracker
  Feed $\mathbf{h}_p^{K_{\tau,l}}$ to posterior knowledge tracker;
  Get posterior knowledge distribution $P(K_{\tau-1} \mid K_{\tau,l}, pos)$;
  Sample $\mathbf{h}_p^{K_{\tau-1,s}}$ from $P(K_{\tau-1} \mid K_{\tau,l}, pos)$;
  Feed $\mathbf{h}_p^{K_{\tau-1,s}}$ to knowledge shifter to get reward;
  Update $\theta_2 = [\theta_{embedding}, \theta_{encoder}, \theta_{pos}]$ using reward;
  ▷ Start reduce the distance between the prior and posterior knowledge distribution $P(K_{\tau-1}|pri)$ and $P(K_{\tau-1}|K_{\tau,j}, pos)$
  Update θ using KLDivLoss and MLE.
end for

3.4.3 Posterior knowledge tracker. Given context representations $\mathbf{h}_p^{X_{\tau-1}}$ and $\mathbf{h}_p^{Y_{\tau-1}}$ and shifted knowledge representation $\mathbf{h}_p^{K_{\tau,j}}$, the posterior knowledge tracker predicts the posterior knowledge tracking distribution $P(K_{\tau-1}|K_{\tau,j}, pos)$ over knowledge pool $K_{\tau-1}$, which is estimated as follows:

$$\begin{aligned}
P(K_{\tau-1}|K_{\tau,j}, pos) &= \text{softmax}(\mathbf{Q}_{pos}\mathbf{K}_{pos}^\top) \in \mathbb{R}^{|K_{\tau-1}|}\\
\mathbf{Q}_{pos} &= \text{mlp}([\mathbf{h}_p^{X_{\tau-1}}; \mathbf{h}_p^{Y_{\tau-1}}; \mathbf{h}_p^{K_{\tau,j}}]) \in \mathbb{R}^{1\times d}\\
\mathbf{K}_{pos} &= \text{mlp}(\mathbf{H}_p^{K_{\tau-1}}) \in \mathbb{R}^{|K_{\tau-1}|\times d}.
\end{aligned} \tag{5}$$

3.5 Decoding layer

We feed $[\mathbf{H}^{X_\tau}; \mathbf{H}^{K_{\tau,j}}]$ into a transformer decoder [38] equipped with a copying mechanism [9, 22, 29] to generate Y_τ token by token, where $\mathbf{H}^{X_\tau} \in \mathbb{R}^{|X_\tau|\times d}$ is the context representation before pooling (see Eq. 2) and $\mathbf{H}^{K_{\tau,j}} \in \mathbb{R}^{|K_{\tau,j}|\times d}$ is the shifted knowledge representation before pooling based on the prediction from the knowledge shifter. Specifically, the probability of generating $y_{\tau,t}$ at t is modeled as:

$$P(y_{\tau,t}) = P(g)P(y_{\tau,t}|g) + P(c_c)P(y_{\tau,t}|c_c) + P(c_k)P(y_{\tau,t}|c_k), \tag{6}$$

where $P(y_{\tau,t}|g)$ is the probability of generating a token from the predefined vocabulary V:

$$P(y_{\tau,t}|g) = \text{mlp}(\mathbf{h}_{\tau,t}) \in \mathbb{R}^{|V|}, \tag{7}$$

where $\mathbf{h}_{\tau,t} = \text{TransformerDecoder}(\mathbf{emb}(y_{\tau,<t}), [\mathbf{H}^{X_\tau}; \mathbf{H}^{K_{\tau,j}}]) \in \mathbb{R}^d$; TransformerDecoder is a stack of Transformer decoder blocks [38]; $\mathbf{emb}(y_{\tau,<t})$ denotes the embedding of $y_{\tau,<t}$.

$P(y_{\tau,t}|c_c)$ is the probability copying of a token from the context X_τ:

$$P(y_{\tau,t}|c_c) = \sum_{i:x_{\tau,i}=y_{\tau,t}} \alpha^c_{\tau,t,i}, \tag{8}$$

where $x_{\tau,i}$ is the i-th token in X_τ and $\alpha^c_{\tau,t,i}$ is the attention distribution on X_τ with $\mathbf{h}_{\tau,t}$ attentively reading $\mathbf{H}^{X_\tau}$ (see Eq.10). $P(y_{\tau,t}|c_k)$ is the probability copying of a token from the knowledge K_τ, j, which is calculated in a similar way. During the training process, we always give the representations of the ground truth shifted knowledge labels $\mathbf{H}^{K_\tau,l}$ to the decoder, where l refer to the ground truth label.

$P(g)$, $P(c_c)$, and $P(c_k)$ are the coordination probabilities among the above three modes: g, c_c and c_k, which are estimated as follows:

$$[P(g), P(c_c), P(c_k)] = \text{softmax}(\text{mlp}([\mathbf{h}_{\tau,t}; \mathbf{c}^c_{\tau,t}; \mathbf{c}^k_{\tau,t}])) \in \mathbb{R}^3, \quad (9)$$

where $\mathbf{c}^c_{\tau,t}$ and $\mathbf{c}^k_{\tau,t}$ are attention vectors derived from $\mathbf{h}_{\tau,t}$ attending to $\mathbf{H}^{X_\tau}$ and $\mathbf{H}^{K_\tau,j}$, respectively. Finally, $\mathbf{c}^c_{\tau,t}$ is calculated as follows:

$$\begin{aligned} \mathbf{c}^c_{\tau,t} &= \alpha^c_{\tau,t}\mathbf{H}^{X_\tau} \in \mathbb{R}^{1\times d} \\ \alpha^c_{\tau,t} &= \text{softmax}(\mathbf{Q}_c\mathbf{K}_c^\top) \in \mathbb{R}^{|X_\tau|} \\ \mathbf{Q}_c &= \text{mlp}(\mathbf{h}_{\tau,t}) \in \mathbb{R}^{1\times d},\ \mathbf{K}_c = \text{mlp}(\mathbf{H}^{X_\tau}) \in \mathbb{R}^{|X_\tau|\times d}. \end{aligned} \quad (10)$$

And $\mathbf{c}^k_{\tau,t}$ is calculated in a similar way.

3.6 Dual knowledge interaction learning

We devise a DukeL scheme to learn DukeNet, which can be divided into two phases: *warm-up training phase* and *dual interaction training phase*, as shown in Algorithm 1.

3.6.1 Warm-up training phase. We first employ the commonly used MLE loss to maximize the likelihood of the demonstrated examples in the training set [12]:

$$\begin{aligned} \mathcal{L}_{pri}(\theta) &= -\log P(K_{\tau-1,l} \mid pri) \\ \mathcal{L}_{pos}(\theta) &= -\log P(K_{\tau-1,l} \mid K_{\tau,l}, pos) \\ \mathcal{L}_{shi}(\theta) &= -\log P(K_{\tau,l} \mid K_{\tau-1,l}, shi) \\ \mathcal{L}_{g}(\theta) &= -\sum_{t=1}^{|Y_\tau|} \log P(y_{\tau,t} \mid y_{<\tau,t}, X_\tau, K_{\tau,l}), \end{aligned} \quad (11)$$

where θ are all the parameters of DukeNet and l refer to the ground truth label. $\mathcal{L}_{pri}(\theta)$ is the prior tracking loss; $\mathcal{L}_{pos}(\theta)$ is the posterior tracking loss; $\mathcal{L}_{shi}(\theta)$ is the shifting loss; and $\mathcal{L}_{g}(\theta)$ is the generation loss. The *final loss* is a linear combination of the four functions just defined:

$$\mathcal{L}(\theta) = \mathcal{L}_{pri}(\theta) + \mathcal{L}_{pos}(\theta) + \mathcal{L}_{shi}(\theta) + \mathcal{L}_{g}(\theta). \quad (12)$$

3.6.2 Dual interaction training phase. For each iteration, given an example sampled from the training set, we first optimize the knowledge shifter. We feed the representation of the ground truth tracked knowledge $\mathbf{h}_p^{K_{\tau-1,l}}$ to the knowledge shifter to get the knowledge shifting distribution $P(K_\tau|K_{\tau-1,l}, shi)$ (Line 5–6 in Algorithm 1). Then we sample knowledge $K_{\tau,s}$ from $P(K_\tau|K_{\tau-1,l}, shi)$ and feed its representation $\mathbf{h}_p^{K_\tau,s}$ to the posterior knowledge tracker to get the probability of recovering the ground truth tracked knowledge label $K_{\tau-1,l}$ (Line 7–8 in Algorithm 1). We regard this recovering probability as a reward:

$$\begin{aligned} \mathbb{E}[R] &= \mathbb{E}[R \log P(K_{\tau,s} \mid K_{\tau-1,l}, shi)] \\ R &= \log[P(K_{\tau-1,l} \mid K_{\tau,s}, pos)], \end{aligned} \quad (13)$$

where R is the reward of the sampled knowledge $K_{\tau,s}$. After that, we use policy gradients [35] to maximize $\mathbb{E}[R]$ and compute the gradient for the parameters $\theta_1 = [\theta_{embedding}, \theta_{encoder}, \theta_{shi}]$ (Line 9 in Algorithm 1):

$$\nabla_{\theta_1}\mathbb{E}[R] = \mathbb{E}[R\nabla_{\theta_1} \log P(K_{\tau,s} \mid K_{\tau-1,l}, shi)]. \quad (14)$$

We then start to optimize the posterior knowledge tracker (the parameters $\theta_2 = [\theta_{embedding}, \theta_{encoder}, \theta_{pos}]$), which is done in a similar way (Line 11–15 in Algorithm 1). After above dual process, we feed the representation of the ground truth shifted knowledge $\mathbf{h}_p^{K_\tau,l}$ to the optimized posterior knowledge tracker to again predict the posterior knowledge tracking distribution $P(K_{\tau-1}|K_{\tau,l}, pos)$. We then distill the gains from $P(K_{\tau-1}|K_{\tau,l}, pos)$ to $P(K_{\tau-1}|pri)$ via KLDivLoss (Line 17 in Algorithm 1) $\mathcal{L}_{kl}(\theta) =$

$$-\sum_{j=1}^{|K_{\tau-1}|} P(K_{\tau-1,j} \mid K_{\tau,l}, pos) \log \frac{P(K_{\tau-1,j} \mid K_{\tau,l}, pos)}{P(K_{\tau-1,j} \mid pri)}. \quad (15)$$

To reduce the impact of inaccurate reward estimation [19], we combine $\mathcal{L}_{kl}(\theta)$ with the MLE losses $[\mathcal{L}_{pos}(\theta), \mathcal{L}_{shi}(\theta), \mathcal{L}_{g}(\theta)]$ linearly and jointly train them:

$$\mathcal{L}(\theta) = \mathcal{L}_{kl}(\theta) + \lambda[\mathcal{L}_{pos}(\theta) + \mathcal{L}_{shi}(\theta) + \mathcal{L}_{g}(\theta)], \quad (16)$$

where λ is a hyper-parameter to control the effect of MLE. We repeat the above process until convergence.

4 EXPERIMENTS

4.1 Research questions

We aim to answer the following research questions:

(RQ1) What is the performance of DukeNet? Does DukeNet outperform state-of-the-art methods? (See §5.1 and §5.2)

(RQ2) Where does the improvement of DukeNet come from? How do the different components contribute to its performance? (See §6.1)

(RQ3) Does the dual knowledge interaction improve both *knowledge tracking* and *knowledge shifting* as the training goes on? (See §6.2)

(RQ4) Is DukeNet able to generate better responses? Are there any failures? (See §6.3)

4.2 Dataset

Following Kim et al. [12], we evaluate our model on two public KGC benchmark datasets, Wizard of Wikipedia [6] and Holl-E [21]. We follow the settings in the original papers to split the data into training, validation and test.

Wizard of Wikipedia is the largest unstructured KGC dataset that is based on sentences to date. The conversations are conducted between two speakers about some given open-domain topics. The one speaker acts as the wizard (knowledge expert), who can use a retrieval system to acquire knowledge sentences from Wikipedia and chooses any to form a response. The other speaker acts as the apprentice (curious learner), who is eager to talk with the wizard about a topic but does not have access to external knowledge. It contains 18,430, 1,948 and 1,933 conversations for training, validation and test, respectively. The test set is further split into two subsets, Test Seen (965 conversations) and Test Unseen (968 conversations). The former contains conversations on topics overlapping with topics in the training set, and the latter contains conversations on topics never seen in the training and validation set. The average number of sentences in a knowledge pool is 67.57.

Holl-E is a document based dataset, i.e., a single document is given as knowledge per conversation. We strictly follow Kim et al. [12] to modify it into a sentence based dataset. It contains 7,228, 930 and 913 conversations for training, validation and test, respectively. There are also two versions of the test set: one with a single reference and the other with multiple references (more than one ground truth knowledge sentences and corresponding responses for each given conversation context). The average number of sentences in a knowledge pool is 60.5.

4.3 Baselines

We compare DukeNet with state-of-the-art KGC methods in the sentence based setting.

- **Seq2Seq** [34] maps the conversation context into the response with an encoder-decoder framework, which does not use any knowledge information.
- **Transformer** [38] implements an encoder-decoder framework by solely relying on multi-head attention mechanism and dispensing with recurrence, which does not use any knowledge information either.
- **MemNet** [8] combines a Seq2Seq model with an external memory network to store knowledge.
- **TMemNet** [6] combines a transformer model with an external memory network in an end-to-end manner, which further introduces an auxiliary loss to better supervise knowledge selection.
- **PostKS** [15] uses response and conversation context to jointly form a posterior knowledge distribution and regards it as pseudo-labels to supervise KS.
- **SKT** [12] sequentially models the history of KS at previous turns via a sequential latent variable model [31]. In addition, SKT uses BERT to encode conversation context and knowledge and incorporates a copying mechanism [9, 29] to promote response generation. To our knowledge, SKT gets the best performance on the dataset Wizard of Wikipedia [6] at the time of writing.

For a fair comparison, we also report the results of PostKS+BERT and TMemNet+BERT, which also use BERT as encoders.

4.4 Evaluation metrics

We conduct both automatic and human evaluations. For automatic evaluation we follow previous KGC studies [20, 48], and use BLEU [1] [23], METEOR [4], ROUGE-1, ROUGE-2, and ROUGE-L [16] for evaluating response generation. In addition, we report Hit@1 (the top 1 accuracy) to evaluate knowledge selection [12, 18] at each turn. For the evaluation of multiple references in the Holl-E dataset, we follow Moghe et al. [21] and Kim et al. [12]. For evaluating RG, we take the max score between responses generated by models and multiple ground truth responses. For evaluating KS, we regard the knowledge selected by model as correct if it matches any of the ground truth knowledge.

For human evaluation we randomly sample 300 examples from each test set on Amazon Mechanical Turk.[2] For each example, we ask three workers to conduct a pairwise comparison between the knowledge sentences/responses selected/generated by DukeNet and the ones selected/generated by a baseline. Specifically, given the conversation context, the knowledge pool used at the current turn[3], the selected knowledge, as well as the generated responses, each worker needs to give a preference (ties are allowed) in terms of three aspects: (1) *Appropriateness*, i.e., which selected knowledge is more appropriate/relevant w.r.t. the given conversation context; (2) *Informativeness*, i.e., which response looks more informative [12, 18, 49]; (3) *Engagingness*, i.e., which response is better in general [12]. Model names were masked out during evaluation.

4.5 Implementation details

For a fair comparison, we implement all models in our experiments based on the same code framework in PyTorch[4] to ensure that they share the same code apart from the method itself. For models without BERT encoder, the word embedding size and hidden size are all set to 256. For models with BERT encoder, we use BERT-Base pre-trained weights and the hidden size is 768. We use the Adam optimizer [13] ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$) to train all models. In particular, we train DukeNet model for 10 epochs for the warm-up training phase (learning rate 0.0005) and 5 epochs for the dual interaction training phase (learning rate 0.00001). λ in Eq. 16 is set to 0.5. We use gradient clipping with a maximum gradient norm of 2. We use the BERT vocabulary[5] (the size is 30,522) for all models. We train all models on TITAN X (Pascal) GPU. The batch size is chosen from (32, 64, 128) according to the GPU memory. We select the best models based on performance on the validation set.

5 EXPERIMENTAL RESULTS

5.1 Automatic evaluation (RQ1)

We list the results of all methods on both the Wizard of Wikipedia and Holl-E datasets in Tables 1 and 2, respectively. Generally, DukeNet significantly outperforms all baselines on both datasets. From the results, we have three main observations.

First, DukeNet outperforms the strongest baseline SKT by around 1–2% in terms of Hit@1 on both datasets. In particular, the improvement of DukeNet over SKT on the test unseen is 1.65%, while on test seen it is 1.28%, which means that DukeNet can better handle unseen cases. The reason is that SKT only models the unidirectional interaction from knowledge tracking to shifting and merely uses demonstrated examples in the training set to optimize model via MLE. In contrast, DukeNet benefits from DukeL, which regards knowledge tracking and shifting as dual tasks to let them boost each other. The learning process of DukeL explores extra knowledge besides the demonstrated ground truth in the training set, which improves the generalization ability on unseen cases.

Second, DukeNet outperforms the other baselines that do not explicitly model knowledge tracking and shifting by a large margin

[1] We use multi-bleu.perl https://github.com/google/seq2seq/blob/master/bin/tools/multi-bleu.perl

[2] https://www.mturk.com/

[3] To reduce the burden of workers, we limit the number of knowledge sentences in the knowledge pool to 10, which contains the knowledge selected by the models (the rest are randomly sampled).

[4] https://pytorch.org/

[5] https://github.com/huggingface/transformers

Table 1: Automatic evaluation results on the Wizard of Wikipedia dataset. Bold face indicates the best result in terms of the corresponding metric. Significant improvements over the best baseline results are marked with * (t-test, $p < 0.05$).

Methods	Test Seen (%)						Test Unseen (%)					
	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Hit@1	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Hit@1
Given only context												
Seq2Seq	0.72	11.60	20.77	3.84	16.83	–	0.22	11.60	18.20	2.08	15.06	–
Transformer	0.75	11.72	21.03	3.86	17.16	–	0.21	11.55	18.67	2.13	15.43	–
Given knowledge and context												
MemNet	0.63	12.79	21.71	3.95	17.20	6.16	0.42	10.36	21.54	2.38	16.58	5.25
PostKS	0.81	13.30	22.57	4.14	18.65	6.43	0.58	11.95	21.45	2.51	16.52	5.31
TMemNet	2.11	16.69	23.50	6.11	19.55	22.61	0.71	13.27	20.27	3.43	16.70	12.43
PostKS + BERT	1.30	13.87	23.12	4.28	19.08	7.69	0.68	12.83	21.35	3.03	16.86	7.03
TMemNet + BERT	2.35	16.80	23.54	6.28	19.71	23.54	0.75	13.96	21.02	3.79	17.31	14.45
SKT	2.63	17.59	24.78	6.83	20.61	25.70	1.31	14.73	22.46	3.80	18.50	17.81
DukeNet	**3.32***	**19.32***	**26.03***	**6.97**	**22.77***	**26.98**	**2.49***	**15.81***	**24.08***	**4.65***	**19.53***	**19.46**

Table 2: Automatic evaluation results on the Holl-E dataset.

Methods	Single golden reference (%)						Multiple golden references (%)					
	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Hit@1	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Hit@1
Given only context												
Seq2Seq	5.45	18.19	27.66	9.60	22.94	–	7.59	22.16	30.87	12.86	26.62	–
Transformer	5.44	17.79	27.07	9.59	22.60	–	9.56	21.97	29.57	13.26	26.52	–
Given knowledge and context												
MemNet	5.31	18.91	30.60	9.88	24.71	7.94	7.44	22.32	34.16	12.80	27.57	7.46
PostKS	5.89	19.18	31.53	10.74	24.85	7.80	8.19	22.87	34.52	13.27	28.39	8.42
TMemNet	6.50	20.65	31.58	10.43	24.96	23.14	8.81	24.34	35.42	13.67	28.87	33.41
PostKS + BERT	6.04	22.78	32.03	11.71	25.66	8.70	10.26	24.28	34.23	13.17	28.71	11.62
TMemNet + BERT	7.84	23.40	33.26	13.04	27.33	26.39	10.24	27.52	37.59	15.61	31.25	36.64
SKT	9.49	28.14	35.09	22.22	29.43	28.50	12.71	35.10	41.08	28.71	35.63	38.30
DukeNet	**11.21***	**30.03***	**37.55***	**24.06***	**30.89***	**30.63**	**14.77***	**37.12***	**44.54***	**30.31***	**36.92**	**40.54**

in terms of Hit@1, e.g., MemNet, PostKS and TMemNet. Especially, DukeNet outperforms the most competitive baseline TMemNet + BERT by around 3–4% on both datasets. The improvements show that explicitly modeling knowledge tracking and shifting can better capture the interaction between the knowledge at adjacent turns. In addition, the tracked knowledge can provide extra evidence and clues to infer the shifted knowledge compared to use context only, which narrows the search space for KS. DukeNet also outperforms all baselines, including SKT, in terms of Response Generation (RG), i.e., the BLEU, METEOR, ROUGE-1, ROUGE-2, and ROUGE-L scores are significantly improved. Note that nothing special is proposed for the decoder in DukeNet, so the higher scores on the generation metrics indicate that better KS performance of DukeNet also improves the quality of RG.

Third, interestingly, we found that the improvement from BERT is limited compared to that on QA tasks [5], e.g., BERT based models have been shown to be much more effective and have already outperformed humans on the SQuAD dataset [26]. In contrast, on the Wizard of Wikipedia dataset, BERT only brings limited gains in terms of Hit@1, e.g., 0.92–2.01%/1.26–1.72% for TMemNet and TMemNet+BERT/PostKS and PostKS+BERT. We think the reason is that BERT is pretrained on language modeling tasks, which will hence help improve the language modeling performance mostly. However, on the KGC task, the main bottleneck now is the KS, which BERT can only make a limited contribution.

5.2 Human evaluation (RQ1)

Although automatic evaluation metrics have been shown to be reliable on the KGC task [6, 20], we still conduct human evaluations to confirm the improvement of DukeNet.

We compare DukeNet with the three most competitive baselines TMemNet + BERT, PostKS + BERT, and SKT on the more challenging Wizard of Wikipedia dataset. The results are shown in Table 3. Generally, DukeNet achieves the best performance in terms of all metrics on both datasets. In particular, we find that the performance gaps between DukeNet and the other three baselines are more obvious on the test unseen subset. For instance, the wins of DukeNet over SKT is 0.15 and 0.20 on the test seen and test unseen in terms of Appropriateness, respectively, which is consistent with the automatic evaluation results and again indicates that exploring

Table 3: Human evaluation on the Wizard of Wikipedia dataset. The 2/3 agreement ratio (2 out of 3 workers give the same annotation) is 93.02%, and the 3/3 agreement ratio is 52.21%.

Methods	**Test Seen (%)**									**Test Unseen (%)**								
	Appropriateness			Informativeness			Engagingness			Appropriateness			Informativeness			Engagingness		
	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose	Win	Tie	Lose
DukeNet vs PostKS + BERT	51	46	3	74	23	3	59	36	5	74	25	1	70	26	4	78	21	1
DukeNet vs TMemNet + BERT	37	55	8	19	79	2	48	42	10	45	50	5	23	74	3	38	57	5
DukeNet vs SKT	15	79	6	15	80	5	16	77	7	20	76	4	15	81	4	21	68	11

Table 4: Ablation study on the Wizard of Wikipedia dataset. -DukeL denotes DukeNet without DukeL. -Pri and -Pos denote DukeNet without prior and posterior knowledge tracker, respectively. -KL denotes DukeNet without KLDivLoss in Eq. 15.

Methods	**Test Seen (%)**						**Test Unseen (%)**					
	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Hit@1	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	Hit@1
Full model	**3.32**	**19.32**	**26.03**	**6.97**	**22.77**	**26.98**	**2.49**	**15.81**	**24.08**	**4.65**	**19.53**	**19.46**
-Pri,-KL	2.96	18.03	25.85	6.52	21.41	25.97	1.77	15.03	23.24	4.28	18.85	18.15
-DukeL	2.81	17.46	24.67	6.61	20.65	25.41	1.49	14.61	22.34	3.93	18.39	17.63
-DukeL,-Pri,-Pos,-KL	2.46	16.89	23.74	6.40	19.72	23.33	1.03	13.89	21.28	3.65	17.45	14.66

data besides the ground truth in DukeL can indeed promote the generalization ability. DukeNet is even better than SKT in terms of Informativeness, despite that the fact that they both use a copying mechanism [9] to make use of knowledge during decoding. We think that this is because, in many cases, SKT selects inappropriate knowledge for a given conversation context; the response generated based on this is more likely to be less relevant and will be considered as less informative by workers. DukeNet gets the best score in terms of Engagingness, which shows that the workers prefer the responses from DukeNet in general, mostly because DukeNet selects more appropriate knowledge which will result in more relevant and natural responses.

6 ANALYSIS

6.1 Ablation study (RQ2)

To analyze where the improvements of DukeNet come from, we conduct an ablation study on the Wizard of Wikipedia dataset; see Table 4. Here, we consider three settings. (1) No prior knowledge tracker (i.e., -Pri,-KL in Table 4), i.e. we directly use posterior knowledge tracker during inference, and feed zero vector to replace the shifted knowledge. (2) No DukeL (i.e., -DukeL in Table 4). (3) No knowledge tracking (i.e., -DukeL,-Pri,-Pos,-KL in Table 4).

The results show that all parts are helpful to DukeNet because removing any of them will decrease the results. Without knowledge tracking, the performance of DukeNet drops sharply in terms of all metrics, almost degenerating to TMemNet + BERT. Specifically, it drops around 3–4% in terms of Hit@1, which means that knowledge tracking is essential for KS, and modeling KS only by modeling conversation context is far from enough. Without DukeL, the performance of DukeNet also drops a lot in terms of all metrics, almost degenerating to SKT. It drops by 1.48% and 1.83% in terms of Hit@1 in the test seen and test unseen conditions, respectively. This indicates that DukeL can not only model the dual interaction between knowledge tracking and shifting to improve them jointly but also address the many-to-many mapping phenomenon in KGC

Table 5: The performances of knowledge tracking and shifting during dual interaction training phase on the Wizard of Wikipedia dataset [12]. T-Hit@1 and S-Hit@1 (same with Hit@1 mentioned before) denote the Hit@1 for knowledge tracking and shifting, respectively.

Epoch	**Test Seen (%)**		**Test Unseen (%)**	
	T-Hit@1	S-Hit@1	T-Hit@1	S-Hit@1
0	74.64	25.41	60.55	17.63
1	75.16	25.84	61.07	17.90
2	75.59	26.19	60.77	18.55
3	75.12	25.75	61.86	19.08
4	75.72	26.59	62.09	18.97
5	76.08	26.98	62.13	19.46

via exploring the knowledge that is not limited to ground truth. Without the prior knowledge tracker, we find that the gain from DukeL is very limited, though it still slightly outperforms SKT. It only improves around 0.5% in terms of Hit@1 compared to the case without DukeL. The posterior knowledge tracker without shifted knowledge as input cannot perform well during inference due to the incompatible dual processes between training and inference, which restricts the effect of the knowledge shifter.

6.2 Dual knowledge interaction (RQ3)

To analyze whether dual knowledge interaction improves both knowledge tracking and knowledge shifting as the training process develops, we report the T-Hit@1 and S-Hit@1 on each epoch during the dual interaction training phase (see §3.6.2), respectively. The results on the Wizard of Wikipedia dataset are shown in Table 5.

We see that the performance keeps improving on both the test seen and test unseen for knowledge tracking and shifting, which indicates that knowledge tracking and shifting indeed teach each other during this process. Thus, during inference, the improved

Table 6: Case study. Due to the space limitations, we merge the knowledge pool $K_{\tau-1}$ and K_τ into one pool and only show three knowledge sentences.

	Example 1 (Test seen)	Example 2 (Test unseen)
Context	$X_{\tau-1}$: pizza delivery $Y_{\tau-1}$: for dinner i had pizza delievered to my house by a pizzeria X_τ: love a good pizzaria or restaurant that specializes in pissa	$X_{\tau-1}$: what can you tell me about the story of harry potter? $Y_{\tau-1}$: harry potter is a 7 book fantasy series written by jk rowling X_τ: 7 books ! wow
Knowledge pool	K_1: pizza delivery is a service in which a pizzeria or pizza chain delivers a pizza to a customer . K_2: pizzas may be delivered in pizza boxes or delivery bags , and deliveries are made with either an automobile . K_3: an order is typically made either by telephone or over the internet to the pizza chain, in which the customer can request pizza type , motorized scooter	K_1: harry potter is a series of fantasy novels written by british author j. k. rowling . K_2: the main story arc concerns harry ' s struggle against lord voldemort , a dark wizard who intends to become immortal . K_3: the books have sold more than 500 million copies worldwide , making them the best-selling book series in history . ...
Tracking to shifting	PostKS + BERT: $\varnothing \rightarrow K_1$ ✗ TMemeNet + BERT: $\varnothing \rightarrow K_1$ ✗ SKT: K_2 ✗ $\rightarrow K_2$ ✗ DukeNet: K_1 ✓ $\rightarrow K_3$ ✓	PostKS + BERT$\varnothing \rightarrow K_1$ ✗ TMemeNet + BERT: $\varnothing \rightarrow K_1$ ✗ SKT: K_1 ✓ $\rightarrow K_3$ ✗ DukeNet: K_1 ✓ $\rightarrow K_2$ ✓
Response	PostKS + BERT: i love pizza delivery . i love pizza delivery . TMemeNet + BERT: i can do it with my pizza delivery , but i can pay my pizza at home . SKT: i love frozen pizza ! i love the delivery bagels with either an automobile , or a motorized scooter . DukeNet: i love pizza delivery and i could pay online or online ordering .	PostKS + BERT: i know that harry potter is a series of fantasy novels . TMemeNet + BERT: i 've never read the book, but i know that harry potter is a series of fantasy novels written by british author j. k. rowling . SKT: they have sold more than 500 million copies worldwide . DukeNet: the main story is about harry ' s struggle against lord voldemort .

knowledge tracking can ground tracked knowledge more accurately, and the more accurately tracked knowledge can be fed to the improved knowledge shifting to select more appropriate knowledge, which eventually boosts the quality of RG. Note that knowledge shifting gets more gains than knowledge tracking on both test sets, i.e., 1.57 vs. 1.43 on test seen and 1.83 vs. 1.58 on test unseen. Knowledge tracking is easier than knowledge shifting, so that knowledge tracking can give a more precise reward signal to help optimize knowledge shifting.

6.3 Case study (RQ4)

We randomly select examples from the Wizard of Wikipedia test sets to compare the performance of DukeNet, SKT, TMemNet+BERT and PostKS+BERT in Table 6. We can see that DukeNet conducts knowledge tracking and shifting more precisely, and hence results in more engaging responses. For instance, in Example 1, DukeNet captures the knowledge interaction from the general definition of "Pizza delivery" to its online ordering. In Example 2, DukeNet shifts the topic from "harry potter novel" to its main story content properly. In contrast, in Example 1, SKT messes up the knowledge K_1 and K_2, and makes a mistake by considering K_2 as the tracked knowledge. The possible reason is that SKT does not explicitly distinguish knowledge tracking and shifting, and uses the unidirectional interaction from knowledge tracking to shifting, which cannot utilize the dual interaction between them to further improve both. TMemNet+BERT and PostKS+BERT get the worst performances in the two examples, which again shows that only using context to model KS is far from enough. We also observe a few bad cases of DukeNet. Although DukeNet alleviates the problem of the many-to-many mapping phenomenon to a certain extent, we found that DukeNet is still more likely to make mistakes when given a tracked knowledge sentence that can be mapped to multiple reasonable shifted knowledge ones. This is because the knowledge interaction between two adjacent turns can only provide limited hints, which fails to model the long-term dual interactions between knowledge tracking and shifting.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose DukeNet, which explicitly models knowledge tracking and knowledge shifting as dual tasks to improve performance on the Knowledge Selection (KS) task. We also devise an unsupervised DukeL to explore knowledge beyond the demonstrated ones in the dataset during training for KS. Extensive experiments on two benchmark datasets show that DukeNet achieves new state-of-the-art performance, indicating that DukeNet enhanced by DukeL can select more appropriate knowledge and hence generate more informative and engaging responses.

A limitation of DukeNet is that it only considers the dual knowledge interaction between two adjacent conversation turns. In future work, we plan to extend the dual knowledge interaction to complete conversations, such that we can leverage long-term dual interactions between knowledge tracking and shifting to further improve Knowledge Selection (KS).

CODE

To facilitate reproducibility of the results in this paper, we are sharing the code at http://url.suppressed.for.anonymity.

REFERENCES

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking Clarifying Questions in Open-domain Information-seeking Conversations. In *SIGIR*. 475–484.
[2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder. *arXiv preprint arXiv:1803.11175* (2018).
[3] Shaobo Cui, Rongzhong Lian, Di Jiang, Yuanfeng Song, Siqi Bao, and Yong Jiang. 2019. DAL: Dual Adversarial Learning for Dialogue Generation. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. 11–20.
[4] Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the ninth workshop on statistical machine translation*. 376–380.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.
[6] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *ICLR*.
[7] Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019. Neural Approaches to Conversational AI. *Foundations and Trends in Information Retrieval* 13, 2-3 (2019), 127–298.
[8] 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 *AAAI*. 5110–5117.
[9] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *ACL*. 1631–1640.
[10] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual Learning for Machine Translation. In *NeurIPS*. 820–828.
[11] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in Building Intelligent Open-domain Dialog Systems. *TOIS* (2020).
[12] Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential Latent Knowledge Selection for Knowledge-Grounded Dialogue. In *ICLR*.
[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *ICLR*.
[14] Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental Transformer with Deliberation Decoder for Document Grounded Conversations. In *ACL*. 12–21.
[15] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. In *IJCAI*.
[16] Chin-Yew Lin. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. 74–81.
[17] Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. Knowledge Diffusion for Neural Dialogue Generation. In *ACL*. 1489–1498.
[18] Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. 2019. Knowledge Aware Conversation Generation with Reasoning on Augmented Graph. In *EMNLP*. 1782–1792.
[19] Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer. In *IJCAI*.
[20] Chuan Meng, Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. RefNet: A Reference-aware Network for Background Based Conversation. In *AAAI*.
[21] Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards Exploiting Background Knowledge for Building Conversation Systems. In *EMNLP*. 2322–2332.
[22] Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019. Multi-style Generative Reading Comprehension. In *ACL*. Florence, Italy, 2273–2284.
[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *ACL*. 311–318.
[24] Prasanna Parthasarathi and Joelle Pineau. 2018. Extending Neural Generative Conversational Model using External Knowledge Sources. In *EMNLP*. 690–695.
[25] Lianhui Qin, Michel Galley, Chris Brockett, and Xiaodon Liu. 2019. Conversing by Reading: Contentful Neural Conversation with On-demand Machine Reading. In *ACL*. 5427–5436.
[26] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2383–2392.
[27] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* 7 (2019), 249–266.
[28] Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. Thinking Globally, Acting Locally: Distantly Supervised Global-to-Local Knowledge Selection for Background Based Conversation. In *AAAI*.
[29] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*. 1073–1083.
[30] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-text Conversation. In *ACL*. 1577–1586.
[31] Shiv Shankar and Sunita Sarawagi. 2019. Posterior Attention Models for Sequence to Sequence Learning. In *ICLR*.
[32] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end Memory Networks. In *NeurIPS*. 2440–2448.
[33] Yibo Sun, Duyu Tang, Nan Duan, Shujie Liu, Zhao Yan, Ming Zhou, Yuanhua Lv, Wenpeng Yin, Xiaocheng Feng, Bing Qin, et al. 2019. Joint Learning of Question Answering and Question Generation. *TKDE* (Early access) (2019).
[34] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NeurIPS*. 3104–3112.
[35] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *NeurIPS*. 1057–1063.
[36] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question Answering and Question Generation as Dual Tasks. *arXiv preprint arXiv:1706.02027* (2017).
[37] Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. 2019. DyKgChat: Benchmarking Dialogue Generation Grounding on Dynamic Knowledge Graphs. In *EMNLP*. 1855–1865.
[38] 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 *NeurIPS*. 5998–6008.
[39] Wenjie Wang, Minlie Huang, Xin-Shun Xu, Fumin Shen, and Liqiang Nie. 2018. Chat More: Deepening and Widening the Chatting Topic via A Deep Model. In *SIGIR*. 255–264.
[40] Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Multi-Agent Dual Learning. In *ICLR*.
[41] Yijun Wang, Yingce Xia, Li Zhao, Jiang Bian, Tao Qin, Guiquan Liu, and Tie-Yan Liu. 2018. Dual Transfer Learning for Neural Machine Translation with Marginal Distribution Regularization. In *AAAI*.
[42] Yingce Xia, Jiang Bian, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Dual Inference for Machine Learning.. In *IJCAI*. 3112–3118.
[43] Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. 2017. Dual Supervised Learning. In *ICML*. 3789–3798.
[44] Yingce Xia, Xu Tan, Fei Tian, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2018. Model-level Dual Learning. In *ICML*. 5383–5392.
[45] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. Dualgan: Unsupervised Dual Learning for Image-to-image Translation. In *ICCV*. 2849–2857.
[46] Hainan Zhang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2018. Reinforcing Coherence for Sequence to Sequence Model in Dialogue Generation.. In *IJCAI*. 4567–4573.
[47] Jiayi Zhang, Chongyang Tao, Zhenjing Xu, Qiaojing Xie, Wei Chen, and Rui Yan. 2019. EnsembleGAN: Adversarial Learning for Retrieval-Generation Ensemble Model on Short-Text Conversation. In *SIGIR*. 435–444.
[48] Wen Zheng and Ke Zhou. 2019. Enhancing Conversational Dialogue Models with Grounded Knowledge. In *CIKM*. 709–718.
[49] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense Knowledge Aware Conversation Generation with Graph Attention.. In *IJCAI*. 4623–4629.
[50] Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A Dataset for Document Grounded Conversations. In *EMNLP*. 708–713.