Think Before You Speak: Explicitly Generating Implicit Commonsense Knowledge for Response Generation

Pei Zhou1* Karthik Gopalakrishnan2 Behnam Hedayatnia2 Seokhwan Kim2
Jay Pujara1 Xiang Ren1 Yang Liu2 Dilek Hakkani-Tur2

1 Department of Computer Science, University of Southern California
2 Amazon Alexa AI

{peiz,jpujara,xiangren}@usc.edu,
{karthgop,behnam,seokhwk,yangliud,hakkanit}@amazon.com

Abstract

Implicit knowledge, such as common sense, is key to fluid human conversations. Current neural response generation (RG) models are trained to generate responses directly, omitting unstated implicit knowledge. In this paper, we present Think-Before-Speaking (TBS), a generative approach to first externalize implicit commonsense knowledge (think) and use this knowledge to generate responses (speak). We expect that externalizing implicit knowledge allows more efficient learning, produces more informative responses, and enables more explainable models. We analyze different choices to collect knowledge-aligned dialogues, represent implicit knowledge, and transition between knowledge and dialogues. Empirical results show TBS models outperform end-to-end and knowledge-augmented RG baselines on most automatic metrics and generate more informative, specific, and commonsense-following responses, as evaluated by human annotators. TBS also generates knowledge that makes sense and is relevant to the dialogue around 85% of the time.

1 Introduction

Human communication strives to achieve common ground, consisting of mutual beliefs and common knowledge (Stalnaker, 1978; Clark and Schaefer, 1989). Such common ground depends not only on utterances, but also implicit knowledge. For example, in Figure 1, this common ground includes the relevant implicit background knowledge “rose is a type of flower”. Integrating such common ground in utterances is an implicit process often referred to as knowledge grounding (Clark and Brennan, 1991). Recent state-of-the-art neural response generation (RG) models based on pre-trained language models (LM) mostly produce responses in an end-to-end manner (Vaswani et al., 2017; Zhang et al., 2020a; Lewis et al., 2020), i.e., models are trained to take history and produce a response. Since implicit knowledge is unstated in dialogue history, RG models do not explicitly learn knowledge grounding and may generate uninformative and hallucinated responses (Serban et al., 2017; Welleck et al., 2019; Roller et al., 2021). Knowledge-grounded RG (Ghazvininejad et al., 2018; Dinan et al., 2019; Gopalakrishnan et al., 2019) addresses this issue, however, most approaches require a knowledge base (KB) to retrieve knowledge for RG (Zhou et al., 2018; Zhao et al., 2020; Eric et al., 2021), which may suffer from the limited knowledge coverage of the used KBs. Some work also casts knowledge as a latent factor in generation (Tuan et al., 2020; Xu et al., 2021), which makes it hard to examine the quality of knowledge generation and how exactly RG uses the implicit knowledge, posing interpretability concerns.

We propose Think-Before-Speaking (TBS), an RG framework that trains the RG model to explicitly generate the implicit knowledge and use this knowledge to generate a response, inspired by inquiry-based discovery learning (Bruner, 1961).
We argue that this decomposition brings three major benefits: 1) compared with end-to-end RG, generated knowledge augments and/or constrains RG to produce more informative responses; 2) compared with knowledge-retrieval models, explicitly generating intermediate groundings can potentially generalize to knowledge not included in KBs and synergize with the RG process; 3) explicitly generated implicit knowledge used in RG provides a faithful explanation of the response intent.

This new RG paradigm poses three main challenges: (1) how to identify implicit commonsense knowledge associated with dialogue turns for training the knowledge generation module; (2) how to represent structured knowledge in natural language (NL) for neural generative models; and (3) how to integrate knowledge and dialogues while distinguishing implicit and explicit parts in responses. To collect knowledge associated with each dialogue instance for training the TBS generative model, we propose weak supervision procedures to automatically align knowledge with each dialogue turn, rather than manually collecting human-annotations, which is expensive and unscalable. This is achieved by using ConceptNet (Speer et al., 2017) as our knowledge base and different matching approaches to identify the implicit knowledge. We explore several ways to format knowledge originally represented as structured triples into natural language so that RG models can adapt to the knowledge+response generation task easily. We experiment with structured triples, triples converted to natural language, and a more colloquial question answering format. To ensure a smooth transition between knowledge and dialogues, we consider using special symbols or prompts as separators.

To evaluate the TBS framework, we introduce new evaluation protocols to cover different aspects of the system, including response quality, knowledge quality, and how TBS models leverage generated knowledge. We conduct extensive human evaluations for different variants of our training procedure. Our experimental results show that our models produce more informative, specific, and responses that make more common sense compared to end-to-end RG models and other knowledge-augmented models such as knowledge-selection. Knowledge quality analysis shows that at least 85% of generated knowledge makes sense and is relevant, and the generated novel knowledge (not in ConceptNet) also has high quality. Furthermore, our TBS model even outperforms an RG model that takes in knowledge obtained using ground-truth responses, showing that explicitly generating implicit knowledge is a promising direction for response generation in open domain dialogue systems.

2 Problem Formulation

Our TBS RG paradigm extends the traditional RG setting by incorporating an additional component of implicit knowledge in the generation process to externalize the knowledge grounding step in RG.

2.1 Response Generation

We follow the common dialogue response generation setup (Weizenbaum, 1966; Ritter et al., 2011; Sordoni et al., 2015): given a dialogue history \( H \) (a sequence of dialogue utterances), generate an appropriate response \( R \). Current neural RG models often frame this task as a conditional language modeling problem. Specifically, given a history \( (H) \) consisting of a sequence of \( n \) dialogue turns: \( X_1, X_2, ..., X_n \) (each turn refers to an utterance containing a sequence of \( t_i \) tokens: \( x_{i,1}, x_{i,2}, ..., x_{i,t_i} \)) and a response \( R \) sentence \( Y \) comprised of a sequence of \( m \) tokens \( y_1, y_2, ..., y_m \), RG models aim to learn the conditional probability distribution by training on human dialogues:

\[
P_\theta(R|H) = \prod_{i=1}^{m} P_\theta(y_i|y_{<i}, X_1, ..., X_n). \tag{1}
\]

2.2 Implicit Knowledge Generation

To make the implicit knowledge grounding step explicit, we introduce a new component to RG – implicit knowledge that is conditioned on the dialogue history \( H \). We use \( I \) to denote the implicit knowledge for brevity, which contains multiple natural language (NL) statements \( I = Z_1, Z_2, ... \) (each containing a sequence of tokens: \( z_{i,1}, z_{i,2}, ... \)) expressing commonsense knowledge. For example, in Figure 1, “rose is a type of flower” and “rose is a symbol of love” are two NL statements expressing the implicit commonsense knowledge. To emulate realistic conversation scenario, we also fuse dialogue history \( H \) in traditional RG with implicit knowledge \( I \) for each turn and denote it with \( H' \), i.e. \( H' = X_1, I_1, X_2, I_2, ..., X_n \), where \( I_i \) indicates the implicit knowledge statements for the i-th turn in the dialogue history.

To externalize the knowledge grounding step, inspired by how humans communicate and inquiry-
based learning (Bruner, 1961; Shwartz et al., 2020a), our TBS RG paradigm requires models to first generate implicit knowledge \( I \) conditioned on \( H' \), i.e., \( P_\theta(H_n \mid H') = X_1, I_1, X_2, I_2, \ldots, X_n \).

3 Learning to Generate Implicit Knowledge by Self-Talk

This section introduces our proposed TBS method to train a generative model that can both talk with itself to explicitly generate background commonsense knowledge \( (P_\theta(I \mid H')) \) and then generate response afterwards, \( P_\theta(R \mid H', I) \). Figure 2 illustrates the process to train the TBS models. To pair each dialogue with appropriate implicit knowledge, we first define a matching process and use ConceptNet (Speer et al., 2017) as the implicit knowledge source (Section 3.1). Then, to construct training instances, we face two key method design choices: how to represent knowledge (3.2) and how to connect the knowledge with the dialogue (3.3). Finally, we train TBS RG models to learn \( P_\theta(I \mid H') \) and \( P_\theta(R \mid H', I) \) with the same parameters \( \theta \). The following sections explain these components in details.

3.1 Knowledge-Aligned Dialogues

To train TBS models we need dialogue datasets consisting of a dialogue history, a response, and the knowledge statement connecting them. We focus on two methods that create weakly-supervised knowledge labels for dialogues as they are more scalable and cost less than human annotations.

Hard-Matching The hard-matching process first lemmatizes all the non-stop words in each utterance, then it identifies knowledge triples whose two concepts appear in an utterance and the next turn respectively. This is the same as the filtering process in Zhou et al. (2021a) and is closely related to distant supervision methods for relation extraction (Craven et al., 1999; Mintz et al., 2009). For more details, refer to Appendix A.1.

Soft-Matching Using Embedding Similarity Hard-matching only captures the surface form and neglects many important semantic relations between words. We thus develop a soft-matching procedure using embedding similarity from SentenceBERT (Reimers and Gurevych, 2019) to measure semantic relations between dialogue turns and triples in ConceptNet. Specifically, we first extract candidate triples from ConceptNet with one concept appearing in the \( i^{th} \) turn. Next, we form a query by concatenating the \( i^{th} \) turn and the next \( (i + 1)^{th} \) turn response. Finally, we encode the query and all triple candidates using SentenceBERT and use cosine similarity to find the semantically closest triples as matched knowledge. More details are presented in Appendix A.1.

3.2 Knowledge Representation

Implicit commonsense knowledge \( I \) stored in ConceptNet is in the form of \((subject, relation, object)\) triples, such as \( (\text{rose}, \text{TypeOf}, \text{flower}) \), which is not compatible with RG models, which operate on NL sentences and may not include relation tokens in their trained vocabulary. Here we design two alternatives to represent the grounded knowledge and use the implicit knowledge in Figure 1 as a running example.

Map Relations to Natural Language (NL) To convert ConceptNet triples into NL, we follow a common practice and map every relation \( r \) in the triple to its NL template, and fill in \( s \) and \( o \) in the template (Levy et al., 2017). We use the same mapping as that used in COMET (Bosselut et al., 2019), covering all standard types of relations in ConceptNet. For example, \( \text{rose is a type of flower; rose is a symbol of love.} \)

Information-Seeking Question-Answer Pairs Another format to convert triples to NL sentences is through asking and answering information-seeking questions. Shwartz et al. (2020b) designed templates of information-seeking questions and answers to provide background knowledge for LMs. We adopt a similar strategy and design a template for each relation in ConceptNet. For example, \( \text{What is a type of flower? Rose is a type of flower.} \)

3.3 Knowledge-Dialogue Transition

To help our RG models learn the TBS paradigm and generate outputs structured similarly, i.e., implicit knowledge first and then responses, we need to properly connect knowledge and dialogues in our data. Here we consider two alternatives for creating such a transition.

Special symbols. Following the common practice of separating sequences in neural LMs (Radford et al., 2018; Devlin et al., 2019), we use a
special symbol to serve as the separator. We enclose the implicit knowledge with special symbols “<implicit>” and “</implicit>” and add it between $H'$ and $R$, for example, “<speaker1> I need to buy some flowers for my wife. <implicit> rose is a type of flower </implicit> <speaker2> Perhaps you’d be interested in red roses.”

**Natural language prompts.** More recent work has found that NL prompts help LMs to perform better on various downstream tasks, including natural language generation (NLG) (Brown et al., 2020; Liu et al., 2021; Zheng and Huang, 2021). Here we use the NL prompts to prompt RG models to generate implicit knowledge and responses. We use “The following background knowledge is helpful for generating the response:” to elicit knowledge and “Grounded on the background knowledge, what does the speaker probably say in the next response?” to elicit response.

### 3.4 Model Training

After constructing knowledge-aligned dialogues, each of our data instances is a sequence of tokens with three components: a dialogue history $H'$ fused with potential implicit knowledge after each turn, implicit knowledge (empty or non-empty) $I$, and a response $R$. We split each instance $d(H', R, I) \in D$ to first train the model to generate just the knowledge $I$ based on $H'$, $P_{\theta}(I|H')$, and then train it to generate $R$ based on both $I$ and $H'$, $P_{\theta}(R|H', I)$.

Formally, we follow standard way of modeling $P_{\theta}$ in auto-regressive neural RG models and use Maximum Likelihood Estimation (MLE) to train our model to maximize $P_{\theta}(I|H')$ (knowledge generation KG) by minimizing the conditional negative log-likelihood loss (NLL):

$$
\mathcal{L}_{KG} = -\sum_{i=1}^{m} \log P_{\theta}(Z_i|Z_{<i}, X_1, ..., X_n),
$$

where $Z_i$ is the i-th statement in $I$. And to model $P_{\theta}(R|H', I)$ we minimize:

$$
\mathcal{L}_{RG} = -\sum_{i=1}^{m} \log P_{\theta}(y_i|y_{<i}, X_1, I_1, ..., X_n).
$$

We train one generative model on these losses in one-pass with splitted instances for KG and RG instead of multiple training phases. During inference, we only provide dialogue history as input and the model has to generate knowledge and responses.

### 4 Experiment Setup

#### 4.1 Dataset

We consider dialogues from four datasets: DailyDialog (Li et al., 2017), EmpatheticDialogues (Rashkin et al., 2019), MuTual (Cui et al., 2020), and SocialIAA-prompted CommonsenseDialogues (Zhou et al., 2021a). For training, we use the filtered version of the four datasets from Zhou et al. (2021a), which ensures each dialogue contains at least one commonsense knowledge triple from ConceptNet. In total, the training data contains 31k dialogues with 159k utterances. We reserve 10% of data as a development set for evaluating model training and selecting hyperparameters. Table 1 shows the number of instances resulted from applying our hard- and soft-matching procedures to our training data in order to construct knowledge-aligned dialogues.

For testing dialogues, to not bias our evaluation toward where common sense is crucial in making
we use DialoGPT-medium (Zhang et al., 2020a) as our base model, which is a commonly-used end-to-end RG model. We fine-tune DialoGPT using all of the 159K dialogue instances. We also use DialoGPT to serve as the backbone model and consider three variables in our TBS model configuration introduced from Sections 3.1 to 3.3: hard-matching or soft-matching, special symbol as separator or NL prompt, and triple-converted-NL to represent knowledge or information seeking QA pairs. To justify our choice of using one model to do both KG and RG, we also compare with TBS- Two Model where we train separate models for knowledge generation (KG) and RG using the same training data. Our default model configuration is hard-symbol-NL.

We also compare several knowledge-grounded RG baselines that retrieve external knowledge or generate knowledge with another model. For retrieval, we follow most common approaches in knowledge-selection (Zhao et al., 2017; Wolf et al., 2020; Eric et al., 2021) and train RoBERTa (Liu et al., 2019) to classify triples using our knowledge-aligned data (matched or not matched), and use it to label candidate triples during testing (KS-RoBERTa). For the generative model, we use COMET (Bosselut et al., 2019) as a commonsense knowledge generator (KG-COMET).

Furthermore, we consider RG models that take the hard-matched or soft-matched knowledge obtained from the ground-truth response (Hard-GT and Soft-GT). Note that though there is noise in hard-matching or soft-matching procedure, this setting uses the next turn response and is likely to provide relevant knowledge. Implementation details for all the models are shown in Appendix B.1.

4.3 Evaluation Protocol

Automatic Evaluation We use standard natural language generation metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015) and SkipThoughts (Kiros et al., 2015). We also use GRADE (Huang et al., 2020), a reference-free metric shown to have consistent correlation with human judgements (Yeh et al., 2021) to ensure the validity of experimental results.

Human Evaluation We conduct extensive human evaluation using 300 randomly sampled instances from unseen test dialogues described above. For response quality, we conduct pairwise comparison where we present a dialogue history and two responses made by two different models and ask them to choose one or select “not sure” based on different criteria (Zhou et al., 2018; Zhang et al., 2020b)². We evaluate on six dimensions: which response is more grammatical, coherent, engaging, informative, specific, and makes common sense (Zhang et al., 2020b; Roller et al., 2021).

More details of the instructions for annotators on each dimension with examples are included in Appendix B.2. For knowledge quality, we evaluate the generated knowledge in isolation (“does this knowledge make sense”) and in conjunction with the context for relevance. We perform majority voting per instance using three annotators from Amazon Mechanical Turk (AMT). We use Fleiss’ Kappa (κ) (Fleiss, 1971) to measure agreement among the annotators.

5 Results

By evaluating our TBS model variants with other baselines, we aim to address the following questions: 1) do TBS models produce better responses than standard end-to-end RG models? 2) compared with other approaches to retrieve or generate additional knowledge, is TBS more helpful for RG? 3) do TBS RG models generate knowledge that makes sense and is relevant to the dialogue context? 4) do TBS models faithfully leverage the generated knowledge?

5.1 Performance of Response Generation

Model variant analysis To find the best-performing configuration of our TBS method, we consider alternatives as discussed in Sections 3.1 to 3.3, and conduct 4 pairwise comparisons: soft vs.

²We choose to conduct pairwise comparison since multiple previous work has shown that it produces a more reliable evaluation than directly asking humans to score the response, which is a highly subjective task (Amidei et al., 2019; Callison-Burch et al., 2007; Celikyilmaz et al., 2020)
Table 2: Human evaluation on response quality when comparing different model variants. We show the percentage of times annotators prefer each variant to **TBS-hard-symbol-NL** and ties, i.e., wins/ties%. Bold-faced numbers indicate statistical significance (p < 0.05) improvement.

<table>
<thead>
<tr>
<th>Model Variants</th>
<th>Grammatical (%)</th>
<th>Coherent (%)</th>
<th>Informative (%)</th>
<th>Specific (%)</th>
<th>Common Sense (%)</th>
<th>Avg (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBS-soft-symbol-NL</td>
<td>53.0/10.0%</td>
<td>46.3/8.7%</td>
<td>48.3/5.3%</td>
<td>41.7/20.6%</td>
<td>51.6/7.6%</td>
<td>52.7%</td>
</tr>
<tr>
<td>TBS-hard-prompt-NL</td>
<td>50.3/4%</td>
<td>49.7/3%</td>
<td>47.9/9%</td>
<td>49.4/6%</td>
<td>51/3%</td>
<td>48.3/2.7%</td>
</tr>
<tr>
<td>TBS-hard-symbol-QA</td>
<td>53.6/0.7%</td>
<td>53.6/5.6%</td>
<td>51.3/4.7%</td>
<td>51.3/3.7%</td>
<td>51.5/5%</td>
<td>54.3%</td>
</tr>
</tbody>
</table>

Table 3: Automatic evaluations using multiple metrics on response quality. All models are based on DialoGPT-medium. Bold-faced are the best performance. One “*” indicates statistical significant (p < 0.05 in Wilcoxon signed-rank test) improvement upon the best-performing non-GT baseline and “**” indicates significant improvement upon the GT baselines.

<table>
<thead>
<tr>
<th>Models</th>
<th>GRADE</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>ROUGE-L</th>
<th>CIDEr</th>
<th>SkipThoughts</th>
</tr>
</thead>
<tbody>
<tr>
<td>DialoGPT-ft (Zhang et al., 2020b)</td>
<td>0.703</td>
<td>0.060</td>
<td>0.026</td>
<td>0.013</td>
<td>0.007</td>
<td>0.061</td>
<td>0.076</td>
<td>0.037</td>
</tr>
<tr>
<td>KS-SBERT (Reimers and Gurevych, 2019)</td>
<td>0.640</td>
<td>0.067</td>
<td>0.024</td>
<td>0.011</td>
<td>0.005</td>
<td>0.061</td>
<td>0.066</td>
<td>0.047</td>
</tr>
<tr>
<td>KS-RoBERTa (Eric et al., 2021)</td>
<td>0.651</td>
<td>0.073</td>
<td>0.026</td>
<td>0.011</td>
<td>0.005</td>
<td>0.061</td>
<td>0.069</td>
<td>0.051</td>
</tr>
<tr>
<td>KG-COMET (Bosselut et al., 2019)</td>
<td>0.648</td>
<td>0.080</td>
<td>0.032</td>
<td>0.015</td>
<td>0.007</td>
<td>0.069</td>
<td>0.076</td>
<td>0.069</td>
</tr>
<tr>
<td>TBS Two Model</td>
<td>0.722</td>
<td>0.091*</td>
<td>0.033</td>
<td>0.014</td>
<td>0.006</td>
<td>0.070</td>
<td>0.075</td>
<td>0.054</td>
</tr>
<tr>
<td>TBS</td>
<td>0.739**</td>
<td>0.091*</td>
<td>0.037</td>
<td>0.020</td>
<td>0.012</td>
<td>0.075*</td>
<td>0.084*</td>
<td>0.087*</td>
</tr>
<tr>
<td>Hard-GT</td>
<td>0.702</td>
<td>0.091</td>
<td>0.035</td>
<td>0.017</td>
<td>0.008</td>
<td>0.075</td>
<td>0.084</td>
<td>0.086</td>
</tr>
<tr>
<td>Soft-GT</td>
<td>0.642</td>
<td>0.070</td>
<td>0.024</td>
<td>0.011</td>
<td>0.005</td>
<td>0.063</td>
<td>0.069</td>
<td>0.053</td>
</tr>
</tbody>
</table>

**hard, prompt vs. symbol, and QA vs. relation-converted NL format.** From Table 2, we find that using soft-matching to create knowledge-aligned dialogue dataset produces more grammatical responses and responses that make more common sense, with $\kappa=0.64-0.73$, indicating substantial agreement according to one interpretation from Landis and Koch (1977). Using QA to represent knowledge makes the responses more grammatical, coherent, commonsensical, and also achieves the best performance on average on six dimensions. We also compare results that combine these alternatives, e.g., soft-symbol-QA (due to space constraints, results are shown in Appendix C.1), however, we do not observe significant improvements after combining these alternatives and our best configuration in terms of average improvement is still hard-symbol-QA. We thus use hard-symbol-QA as our final configuration and refer to it as **TBS** throughout this section.

**Does TBS produce better responses vs. end-to-end RG?** By comparing TBS and end-to-end DialoGPT-ft model in Table 3 and Figure 3, we find that TBS models produce better-quality responses using both automatic and human evaluations. Specifically, even though hard-matching only annotates about 33% of the training instances, TBS outperforms end-to-end RG model significantly on most automatic metrics. From human evaluation ($\kappa=0.62-0.69$), we find our TBS model performs on par with DialoGPT trained on more data in grammar, coherence, and engagingness, and achieves statistically-significant (p < 0.05) improvement on informativeness, specificity, and the common sense aspects of generated responses. We argue that by providing weakly-supervised knowledge labels and TBS training, RG models require less data and can generate quality responses with improvement in the informativeness, specificity, and common sense aspects of the responses.

**Is TBS knowledge generation better than other knowledge-augmented RG?** We compare TBS models with other knowledge-augmented baselines that retrieve knowledge from ConceptNet using embedding scores (KS-SBERT) or a trained selector (KS-RoBERTa), or generate from another model (KG-COMET). From Table 3, we find that these models perform similarly to the end-to-end DialoGPT model and are outperformed by TBS models on most automatic metrics. Figure 3 shows that while TBS methods have significant improvements on all dimensions against knowledge-selection baselines, COMET as a knowledge generator has smaller gaps on informativeness, specificity, and common sense, but is outperformed significantly on grammar, coherence, and engagingness.

Next we compare against the setup where we feed the model the knowledge that is derived using the ground-truth response (Hard/Soft-GT), i.e., the provided knowledge is obtained using concepts appearing in the ground-truth response. From Table 3, we surprisingly find that even though our
proposed TBS model has no access to response-leaking knowledge labels and is trained on much less data, the TBS RG model still achieves statistically significant improvement on GRADE and BLEU-4. And from human evaluation results in Figure 4, TBS model significantly improves the specificity and common sense aspect of responses while stays on par on other evaluation dimensions compared with the hard-GT model and improves even more compared with soft-GT. We find that one potential explanation is that only around 55% of Hard-GT knowledge is labeled as used in response whereas it is 77% in our TBS model (see Section 5.3). This is also related to how the RG model leverages the knowledge in training. Further analysis is needed to understand the effect of knowledge and the relationship between knowledge and responses.

5.2 Quality of Generated Knowledge

We then examine how well TBS RG models learn to generate knowledge on unseen dialogues. We use human evaluation and focus on three dimensions: does the model generate novel knowledge that does not appear in ConceptNet? does the generated knowledge statement make sense as a standalone fact? and is the generated knowledge relevant to the dialogue context? For the first question we directly query from ConceptNet and show percentages. For the latter two we follow Section 4.3 and show the percentages that MTurkers think the knowledge makes sense and is relevant from the 300 sampled test instances (the same used in response quality). We test our TBS model, the two-model variant, and other knowledge-augmented baselines introduced in Section 4.2.

Around 85% of knowledge generated from TBS makes sense and is relevant  Table 4 shows that TBS models can generate implicit knowledge that makes sense and is relevant to the context for around 85% of the time as judged by human annotators ($\kappa$=0.73-0.80). Compared with knowledge-selection models that retrieve knowledge from ConceptNet, TBS generates knowledge that is similar in terms of common sense and has better relevance to the dialogue history. Compared with COMET that also generates knowledge, we find TBS models generate more knowledge that follows common sense and is relevant to the dialogue. Comparing two-model and one-model TBS, we find that two-model generates more knowledge that makes sense and is relevant, although its response quality is poorer (Table 3 and Figure 3). This might be due...
to model synergies when learning both knowledge generation and response generation.

**Model generates novel knowledge** We find a significant portion of novel knowledge generated from the COMET and TBS models that is not present in the training data. Furthermore, the quality of the generated novel knowledge is similar to that of knowledge existing in ConceptNet. COMET generates more new knowledge but the quality (both common sense and relevance) is significantly lower than TBS models. We include some examples of novel knowledge generated in Appendix C. In general we find that the new knowledge is complimentary to ConceptNet, not just a paraphrased version of existing triples (since in those cases the model will directly generate the ConceptNet triple). This shows a promising sign that TBS RG models can potentially generate good-quality novel knowledge labels for unseen dialogues.

### 5.3 Performance Analysis

**Most responses are knowledge grounded** To examine how TBS methods leverage knowledge for RG, we also present annotators a history, generated knowledge, and generated response, and ask them whether the knowledge is used in response. We find that around 77% of generated knowledge is used in the generated response, i.e., the response is grounded in the knowledge generated from TBS.

**Noisy knowledge heavily impacts quality** To better showcase the connection between knowledge and response, we examine how knowledge quality generated from TBS methods can affect response quality. During inference, we randomly sample noisy knowledge from another dialogue, feed it to the model to generate a response conditioned on irrelevant knowledge, and compare the response quality with response generated from TBS knowledge. Fig 5 shows that there is a statistically significant (p ≤ 0.05) drop in response quality in four dimensions. This indicates that the quality of knowledge input heavily influences response quality and that TBS models generate better responses because of its decent knowledge quality.

### Qualitative examples and limitations

We show several qualitative examples from different models and human responses in Table 5. We find that TBS generates relevant knowledge and responses grounded properly in that knowledge, whereas KS/KG models retrieve noisy knowledge and HardGT generates response not grounded in knowledge.

Here we present a summary of error patterns of TBS models and discuss potential directions to improve. More examples can be found in Table 6. First, our matching procedures do not concern multi-hop triples that might be needed for complex reasoning chains. Second, ConceptNet mostly contains taxonomic and lexical knowledge ("RelatedTo, IsA, etc"), limiting the diversity of generated knowledge from TBS models. We plan to explore other knowledge resources such as ATOMIC2020 (Hwang et al., 2021) in the future.

Third, currently the model always generates implicit knowledge. In future work, we are interested in training RG models that understand when implicit knowledge is needed based on the dialogue context.

### 6 Related Work

**Open-Domain Dialogue Generation** Recent work focused on fine-tuning large pre-trained transformer models (Radford et al., 2019; Zhang et al., 2020a; Roller et al., 2021) on massive dialogue data. Knowledge-augmented RG has been studied extensively to alleviate the issue of generic or hallucinated responses (Serban et al., 2017; Welleck et al., 2019; Roller et al., 2021). Most work retrieves relevant knowledge from knowledge candidates (wikipedia or KBs) and generates responses after incorporating additional knowledge in dialogue context (Ghazvininejad et al., 2018; Zhou et al., 2018; Wu et al., 2020).

More recent work also explored other ways of constructing knowledge, such as by considering knowledge as a latent variable (Tuan et al., 2020; Li et al., 2020) and generating it implicitly. Our TBS framework differs from these two lines of work in that it explicitly generates knowledge in text and uses one generative model for both knowledge generation and RG.
### Table 5: Randomly sampled qualitative examples from our TBS model (TBS-hard-symbol-QA).

<table>
<thead>
<tr>
<th>Dialogue History</th>
<th>Generated Knowledge</th>
<th>Generated Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>B: Wow that is a long time! Hopefully you can stay in touch with all of your former coworkers.</td>
<td>What is school related to? school is related to college; what is job related to? job is related to work; what is pay related to? pay is related to job.</td>
<td>B: I'm sure you'll do great. I'm sure you will. I'm not sure where I want to live.</td>
</tr>
<tr>
<td>B: I'm sorry to hear that. I hope you have a good college experience!</td>
<td>your is related to being; angry is a synonym of upset; cry is related to upset; move is related to leave; move is related to change.</td>
<td>B: Oh wow! Will they pay you a lot?</td>
</tr>
<tr>
<td>B: I'm sorry to hear that. I hope you have a good college experience!</td>
<td>your is related to being; angry is a synonym of upset; cry is related to upset; move is related to leave; move is related to change.</td>
<td>B: You're not alone in that. I'm sorry you had to go through that.</td>
</tr>
<tr>
<td>I got a new job.</td>
<td>What happened? Why?</td>
<td>B: I'm sure you wish; I'm sure you're able to do a new job.</td>
</tr>
<tr>
<td>I am moving out of state.</td>
<td>What is move related to? move is related to change; you is related to young; long is related to tall.</td>
<td>B: I'm so sorry! I hope you are able to get a new job!</td>
</tr>
</tbody>
</table>

### Table 6: Randomly sampled qualitative examples from TBS-hard-symbol-QA. We can find that in most cases the generated knowledge makes sense and responses are grounded in the knowledge and not generic. However, we do find some error cases such as the last example the response does not use the knowledge.

<table>
<thead>
<tr>
<th>Dialogue History</th>
<th>Generated Knowledge</th>
<th>Generated Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>I got so emotional driving home and I cried.</td>
<td>What is job related to? job is related to work; what is pay related to? pay is related to job.</td>
<td>B: I'm sorry to hear that. I hope you find a better job.</td>
</tr>
<tr>
<td>I was there for more than 10 years. I will miss everyone.</td>
<td>What is man related to? man is related to woman; what is happy related to? happy is related to special; what is move related to? move is related to change.</td>
<td>B: Im going back to school full time. I was there for more than 10 years. I will miss everyone.</td>
</tr>
<tr>
<td>I'm calling friends to let you know that I am moving out of state.</td>
<td>What is school related to? school is related to college; what is job related to? job is related to work; what is pay related to? pay is related to job.</td>
<td>B: I'm going back to school full time. I was there for more than 10 years. I will miss everyone.</td>
</tr>
<tr>
<td>I think my parents are not trying to understand.</td>
<td>What is man related to? man is related to woman; what is happy related to? happy is related to special; what is move related to? move is related to change.</td>
<td>B: I'm so sorry! I hope you are able to get a new job!</td>
</tr>
<tr>
<td>I was so angry and hurt, I thought I was going to be physically ill or something.</td>
<td>What is man related to? man is related to young; what is young? young is a man; what is man related to? man is related to woman.</td>
<td>B: I'm sorry to hear that. I hope you find a better job.</td>
</tr>
</tbody>
</table>

### Generating Knowledge for Natural Language Understanding (NLU) Although explicit knowledge generation (KG) for RG has not been explored, similar methods have been proposed for NLU tasks such as question answering (Shwartz et al., 2020b). Previous work has also explicitly generated rationales that can be seen as helpful additional knowledge (Rajani et al., 2019). TBS differs from such work in that we consider a generative task and use the same generative model to do both KG and RG.

### 7 Conclusion

Inspired by how humans contribute to the common ground during communication, we propose to train RG models that explicitly generate implicit knowledge and then respond (TBS). This brings us three main benefits compared with prior end-to-end RG models: 1) more informative and coherent responses by augmenting with knowledge; 2) generated knowledge provides faithful explanations of RG model’s inner-workings; 3) models do not rely on external knowledge bases in response generation time. We first identify implicit knowledge in dialogues, explore different knowledge representation and transition choices, and demonstrate promising results compared with end-to-end and knowledge-grounded RG models from extensive evaluations. We find strong and promising results for TBS RG model compared with end-to-end RG. In particular, TBS can produce good quality and novel knowledge, outperform end-to-end RG models despite training on less data, and even produce better responses than RG models that take ground-truth knowledge. We hope our findings encourage more future studies on making RG models better emulate human communication process and produce better-quality responses.

### Ethics and Broader Impact

Our work aims to train RG models that explicitly generate implicit knowledge before responding. Sheng et al. (2021) have found biases in DialoGPT (our base model) responses and Mehrabi et al. (2021) have found representational harms in common sense resources. We acknowledge that the
generated responses from our models might contain biases. All of the dialogue datasets and models are in English, which benefits English speakers more. We have conducted human evaluation using Amazon Mechanical Turks. We pay turkers around $15 per hour, well above the highest state minimum wage and engage in constructive discussions if they have concerns about the process. We also give each annotation instance enough time so that we do not pressure annotators.

Acknowledgments

We thank anonymous reviewers for providing insightful feedback and members from Amazon Alexa AI team and INK and JAUNTS lab from USC. Pei Zhou, Jay Pujara, and Xiang Ren’s work on this project was funded by the Defense Advanced Research Projects Agency with award N660011924033. The research was also supported by gifts from Google.

References


Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit. “ O’Reilly Media, Inc.”.


(AAAI-18), the 30th innovative Applications of Artificial Intelligence (AAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018, pages 5110–5117. AAAI Press.


Ninareh Mehrabi, Pei Zhou, Fred Morstatter, Jay Pujara, Xiang Ren, and Aram Galstyan. 2021. Lawyers are dishonest? quantifying representational harms in commonsense knowledge resources. In EMNLP.


A TBS Framework Details

A.1 Matching Detail

Hard-Matching  This process follows that used in Zhou et al. (2021a). We first identify potential candidates for concepts in ConceptNet (Speer et al., 2017). For each utterance, we use a part-of-speech (POS) tagger to find the nouns, verbs, and adjectives that are not stopwords and then construct a set of potential concepts by including the lemmatized version of these words. The POS tagger, lemmatizer, and stopword list are from the Natural Language Toolkit (NLTK) package (Bird et al., 2009). This step results in a set of concept words for each turn of a dialogue.

With a set of concepts we extract for every dialogue turn, we then identify a list of candidate triples $(e_1, r, e_2)$. We use the ConceptNet containing single-word concepts pre-processed by Zhou et al. (2018). For each concept we identified in a turn, we store all triples in ConceptNet that contain this concept, either as subject or object.

After getting a list of commonsense triples $(e_1, r, e_2)$ containing concepts in a particular turn using ConceptNet, we next examine if any of the other entity in the triples appears in the concept set of the next turn. If we find such a match, we record this triple to be a commonsense assertion that might be implied in the response.

Soft-Matching  We reuse the first several steps of hard-matching to find a set of candidate triples for each dialogue turn, then instead of searching for the exact words in the next turn, we use embedding similarity from SentenceBERT (Reimers and Gurevych, 2019) (specifically the “all-MiniLM-L6-v2” variant, which is claimed to be a “All-round model tuned for many use-cases. Trained on a large and diverse dataset of over 1 billion training pairs”)4.

To select the final matched knowledge, we choose the top 3 triples from ConceptNet with the highest similarity. After examining the distribution of embedding similarities from SBERT, we also require the similarity to be above 0.4 to be matched to ensure quality matching.

A.2 Mappings

We show complete mappings of relations from ConceptNet for both relation-converted NL and information-seeking QA pairs in Table 7.

B Experimental Details

B.1 Implementation Details

We use base models from HuggingFace5 and implement TBS based on TransferTransfo (Wolf et al., 2019)6. We fine-tune the model for 3 epochs with batch size 4 and set the learning rate to be 6.25e-5. We perform gradient accumulation for 8 steps and gradient clipping with a max norm of 1.0 and optimize using the Adam optimizer. For decoding, we use top-p nucleus sampling (Holtzman et al., 2019) with temperature T (p = 0.9 and T = 0.7), and a maximum decoding length of 300 tokens. Note that since we are also generating knowledge, this maximum length is larger than normal RG models. Our TBS models are mostly trained on 4 Quadro RTX 8000 GPUs and take around 5 hours. For automatic metrics, we use the nlg-eval package7 and the GRADE repo8.

B.2 Evaluation Detail

We present the MTurk interface we use for response quality and knowledge quality evaluation in Figures 7, 8, and 9 including instructions and examples. We require turkers to have at least 500 numbers of HITs approved, with approval rate higher than 95%, and from either Canada, UK, or US since our data is in English.
Table 7: Knowledge representation mappings.

### C. Additional Results

#### C.1 Models Combining Variants

Table 8 presents the complete results considering all of our models’ variants. We find that the best overall configuration is hard-symbol-QA.

#### C.2 CEDAR Probing: Do TBS models understand why a response makes sense?

We follow the CEDAR probing framework from Zhou et al. (2021b) that analyzes if RG models assign a higher probability to the response when provided with valid common sense in the form of explanations compared to corrupted explanations. Results comparing to an end-to-end RG model and a knowledge-selection model are shown in Table 9. We find that by TBS training, RG models become much more sensitive to commonsense explanations against complete corruptions but still fall short against more subtle logical corruptions that require deeper reasoning.

---

© DialogPT-medium: [https://huggingface.co/microsoft/DialoGPT-medium](https://huggingface.co/microsoft/DialoGPT-medium)

© transfer-learning-conv-ai: [https://github.com/huggingface/transfer-learning-conv-ai](https://github.com/huggingface/transfer-learning-conv-ai)

© Maluuba/nlg-eval: [https://github.com/Maluuba/nlg-eval](https://github.com/Maluuba/nlg-eval)

© GRADE: [https://github.com/li3cmz/GRADE](https://github.com/li3cmz/GRADE)
Figure 7: Human evaluation interface for response quality on dimensions: grammar, coherence, and engagingness.

Figure 8: Human evaluation interface for response quality on dimensions: informativeness, specificity, and common sense.

Figure 9: Human evaluation interface for knowledge quality with 3 questions: does the knowledge make sense as a standalone fact, is the knowledge relevant to the context, and does the generated response use the knowledge?

Table 8: Human evaluation on response quality when comparing different model variants with the base model (hard-symbol-NL).

Table 9: CEDAR (Zhou et al., 2021b) results where bold-faced numbers indicate statistically significant differences comparing to the second-best model.