Supplementary Material of "Learning to Express in Knowledge-Grounded Conversation"

Anonymous Author(s)
Affiliation
Address
email

1 Derivation of ELBO

$$
\begin{aligned}
&\log p(R|U,K) \\
&= \log \sum_{(M,Z)} p(R,M,Z) \\
&= \log \sum_{(M,Z)} q(M,Z|R) \frac{p(R,M,Z)}{q(M,Z|R)} \\
&= \log \mathbb{E}_{(M,Z)\sim q(M,Z|R)} \frac{p(R,M,Z)}{q(M,Z|R)} \\
&\geq \mathbb{E}_{(M,Z)\sim q(M,Z|R)} \log \frac{p(R,M,Z)}{q(M,Z|R)} \\
&= \mathbb{E}_{(M,Z)\sim q(M,Z|R)} \log p(R|M,Z) - \mathbb{E}_{(M,Z)\sim q(M,Z|R)} \big(\log q(M,Z|R) - \log p(M,Z) \big).
\end{aligned}
\tag{1}
$$

According to the mean-filed approximation, $q(M,Z) \approx q(M)q(Z)$. Therefore, $\mathbb{E}_{(M,Z)\sim q(M,Z|R)} \log p(R|M,Z)$ and $\mathbb{E}_{(M,Z)\sim q(M,Z|R)} \big(\log q(M,Z|R) - \log p(M,Z) \big)$ can be re-written as:

$$
\begin{aligned}
&\mathbb{E}_{(M,Z)\sim q(M,Z|R)} \log p(R|M,Z) \\
&= \mathbb{E}_{M\sim q(M|R)} \big(\mathbb{E}_{Z\sim q(Z|M,R)} \sum_{t=1}^{l_r} \log p(r_t|r_{<t}, z_t)) \big)
\end{aligned}
\tag{2}
$$

$$
\begin{aligned}
&\mathbb{E}_{(M,Z)\sim q(M,Z|R)} \big(\log q(M,Z|R) - \log p(M,Z) \big) \\
&= \mathbb{E}_{M\sim q(M|R)} \Bigg(\mathbb{E}_{Z\sim q(Z|M,R)} \big(\log q(M|R) - \log p(M) \big) + \mathbb{E}_{Z\sim q(Z|M,R)} \big(\log q(Z|M,R) - \log p(Z) \big) \Bigg) \\
&= \mathbb{E}_{M\sim q(M|R)} \Bigg(\log q(M|R) - \log p(M) \Bigg) + \mathbb{E}_{M\sim q(M|R)} \Bigg(\mathbb{E}_{Z\sim q(Z|M,R)} \big(\log q(Z|M,R) - \log p(Z) \big) \Bigg) \\
&= \sum_{t=1}^{l_r} \Bigg(\mathbb{E}_{M\sim q(M|R)} \big(\log q(m_t) - \log p(m_t) \big) \Bigg) + \mathbb{E}_{M\sim q(M|R)} \Bigg(\sum_{t=1}^{l_r} m_{t-1} \cdot \mathbb{E}_{Z\sim q(Z|M,R)} \big(\log q(z_t) - \log p(z_t) \big) \Bigg) \\
&= \sum_{t=1}^{l_r} D_{\mathrm{KL}}(q(m_t) \| p(m_t)) + \mathbb{E}_{M\sim q(M|R)} \Bigg(\sum_{t=1}^{l_r} m_{t-1} \cdot D_{\mathrm{KL}}(q(z_t) \| p(z_t)) \Bigg).
\end{aligned}
\tag{3}
$$

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

2 Details about the Construction of $\tilde{\mathrm{M}}$ and $\tilde{\mathrm{Z}}$

In this section, we provide more details about of construction of $\tilde{M}$ and $\tilde{Z}$. For every response in the training set, we parse it as a syntax tree using StanfordNLP toolkit [7]. The syntax tree we obtain is in a hierarchical and nested structure. The root node of the tree represents the whole response sentence and the root node of every subtree represents a corresponding phrase, a small part of a sentence. For example, if a phrase could be divided into three parts, then the node representing the phrase has three child nodes and each represents a part of the phrase. After we acquire the parsing tree, segmentation is then carried out recursively. To be concrete, we traverse the parsing tree by deep-first search order. Every time we arrive at a node, compute the similarity[1] between the knowledge and the phrase represented by the node. If the similarity is above the threshold μ_{seg}, we mark the phrase as a segment and search in this branch terminates. Else we continue to search the child nodes of the current node to segment at a more refined level. We use $\tilde{M} = \{\tilde{m}_t\}_{t=1}^{l_r}$ to denote the results of segmentation labeling.

The pseudo label of module choice $\tilde{Z} = \{z_t\}_{t=1}^{l_r}$ is tagged in a similar way to multiclass classification. Specifically, for a segment $(r_s, \cdots, r_e)$ where s and e are the start and end position of a segment respectively. If the similarity between this segment and the knowledge falls below a threshold μ_{knl}, its pseudo label $(z_s, \cdots, z_e)$ will be set to 0. Otherwise we send the segment to a series of style discriminators one after another until the classification confidence given by a discriminator is above μ_{sty_i} and pseudo module choice label will be set to $i+1$. If all discriminators fail to classify the segment at a confidence greater than μ_{sty_i}, $(z_s, \cdots, z_e)$ are all 1, indicating knowledge should be expressed without particular style.

3 Learning Algorithm

The learning algorithm is summarized in Algorithm 1.

Algorithm 1 Learning Algorithm

Input: Training data $\mathcal{D}$, thresholds for weak supervision μ_{seg}, μ_{knl} and μ_{sty}, discriminator$\{Dis_i\}_{i=1}^{N_{sty}}$, maximum step M, adapter training step $M^{'}$.
for $m \leftarrow 1$ to M **do**
    Sample a mini-batch $\{(U_i, K_i, R_i)\}$ from $\mathcal{D}$.
    Conduct segmentation on R_i to get $\tilde{M}$.
    for $i \leftarrow 1$ to N_{seg} **do**
        for $j \leftarrow 1$ to N_{sty} **do**
            Use Dis_j to classify response segment $(r_{s_i}, \cdots, r_{e_i})$.
            if Confidence of $Dis_j \geq \mu_{sty}$ and $(z_{s_i}, \cdots, z_{e_i})$ are not assigned **then**
                $(z_{s_i}, \cdots, z_{e_i}) \leftarrow j+1$
            end if
        end for
    end for
    if $m \leq M'$ **then**
        Update the adapters based on the first term in ELBO.
    else
        Update the parameters θ (i.e., θ_m, θ_z and the parameters in $p(r_t)$) and ϕ (i.e., ϕ_m and ϕ_z) based on ELBO and Weak Supervision.
    end if
end for
return Generation Model $p_\theta(R|U,K)$ with prior distribution p_{θ_m} and p_{θ_z}

4 More Implementation Details

We employ a knowledge selection(KS) module to select the top 7 related sentences in knowledge. The KS module is implemented based on Roberta-base(125M) and trained on the Reddit Corpus. Specifically, we treat the sentence which has the highest F1 score with the response as the positive

[1]We use unigram Precision to calculate the similarity.

sample, and the negative sample is randomly sampled from all the other knowledge sentences. We train the KS module via maximum likelihood estimation (MLE) with a batch size of 64 and an initial learning rate of $1e-5$. The threshold μ_{seg}, μ_{knl}, μ_{pos} and μ_{neg}[2] in weak supervision are set as 0.9, 0.5, 0.8 and 0.8, respectively. The encoder-decoder architecture is implemented on the basis of Bart-base(139M) and trained on the Reddit Corpus with a batch size of 64 and an initial learning rate of $5e-6$. The parameters for prior and posterior distributions of Z and M(i.e., θ_z, θ_m, ϕ_z and ϕ_m) are initialized randomly, and optimized with a learning rate of $1e-4$. The parameters for adapters are initialized randomly and optimized with a learning rate of $2e-3$. We only train the adapters for the first 1000 steps. We utilize gated recurrent units (GRUs) as the basic units in $f_{z\text{-rnn}}$. We set the hidden size and the number of layers of RNN in our model(i.e., $f_{z\text{-rnn}}$ and $\psi(\cdot)$) as 128 and 1 respectively. The embedding size for Z is set as 128 and the adapter size is set as 64. When fine-tuning the model on the Wizard and CMU_DoG datasets, the learning rate and the batch size are set as $5e-5$ and 32 respectively. We employ greedy search in response decoding. All models are learned with Adam [4] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We increase the learning rate linearly for the first 200 steps and decrease it thereafter proportionally to the inverse square root of the step number. Early stopping on validation is adopted as a regularization strategy. All models are trained on a 8×RTX 2080 Ti machine.

5 Details of Datasets

Table 1: Statistics of the two datasets.

	Wizard of Wikipedia				CMU_DoG		
	Train	Valid	Test Seen	Test Unseen	Train	Valid	Test
# Utterances	166,787	17,715	8,715	8,782	74,717	4,993	13,646
# Conversations	18,430	1,948	965	968	3,373	229	619
# Topics/Documents	1,247	599	533	58	30	30	30
Avg. # of Turns	9.0	9.1	9.0	9.1	22.2	21.8	22.0

Table 1 reports the statistics of the Wizard data and the CMU_DoG data

6 Comparison with More Baselines

Table 2: Automatic evaluation results.

Training Data	Models	Wizard Seen			Wizard Unseen			CMU_DoG		
		F1	D-1	D-2	F1	D-1	D-2	F1	D-1	D-2
100% annotated data	TMN[1]	15.9	0.041	0.176	14.3	0.025	0.106	9.9	0.003	0.008
	SKT[3]	19.3	0.085	0.300	16.1	0.056	0.188	-	-	-
	DRD[8]	19.3	0.065	0.252	17.9	0.046	0.177	10.7	0.010	0.044
	KnowledGPT[9]	22.0	0.141	0.431	20.5	0.094	0.290	13.5	0.023	0.113
Reddit Corpus	BART[5]	18.4	0.076	0.355	18.4	0.049	0.237	9.8	0.021	0.131
	ZRKGC[6]	18.9	0.055	0.246	18.8	0.037	0.179	12.2	0.015	0.094
	Our Model	19.3	0.082	0.383	19.2	0.060	0.292	12.2	0.028	0.186
Reddit Corpus + 10% annotated data	Our Model	20.4	0.073	0.366	20.0	0.052	0.270	14.4	0.015	0.122

We compare with models trained on full training data, and Table 2 shows the evaluation results. First, it is noted that our model outperforms KnowledGPT in terms of F1 by using only 10% training data[3] on CMU_DoG, which provides a strong support for the effectiveness of the proposed model. Second, by adjusting the structure style on a small amount of data, the gap between our model and KnowledGPT is further narrowed, while the improvement on ZRKGC and BART is trivial.

[2]We consider positive and negative sentiment style in our experiments.

[3]The 10% training data is randomly sampled. The result is an average value of three repetitive experiments on every dataset

7 Ablation over Weak Supervision

Table 3: Ablation study over the weak supervision.

Training Data	Models	Wizard Seen			Wizard Unseen			CMU_DoG		
		F1	D-1	D-2	F1	D-1	D-2	F1	D-1	D-2
Reddit Corpus	Our model	19.3	0.082	0.383	19.2	0.060	0.292	12.2	0.028	0.186
	-weak supervision on Z	19.1	0.077	0.362	19.1	0.056	0.270	10.2	0.027	0.155
	-weak supervision on Z and M	19.1	0.083	0.382	18.8	0.058	0.270	9.5	0.023	0.147
Reddit Corpus + 10% annotated data	Our model	20.4	0.073	0.366	20.0	0.052	0.270	14.4	0.015	0.122
	-weak supervision on Z	19.5	0.072	0.354	19.3	0.051	0.250	13.2	0.014	0.115
	-weak supervision on Z and M	19.5	0.077	0.366	19.2	0.054	0.258	13.5	0.013	0.091

To have more insights into the impact of weak supervision on the performance of our model, we compare the proposed model with the following variants: (1)-*weak supervision on Z*: the weak supervision on module indicator Z is removed; (2)-*weak supervision on Z and M*: the weak supervision on module indicator and boundary indicator is removed. Table 3 reports the evaluation results. We can conclude that (1) the weak supervision objectives significantly improve model performance; (2) the weak supervision objectives play a more crucial role on CMU_DoG, as removing them causes a dramatic drop in performance. The reason is that this dataset has more sophisticated expression styles and it is difficult to learn these styles without auxiliary supervision signals.

8 Human Evaluation

To further verify whether our model could learn structure style and content style, we randomly sample 200 examples from Test Seen of Wizard, and the test set of CMU_DoG respectively, and recruit 3 well-educated native speakers to do qualitative analysis on the responses generated by our model and all baselines. For each of the 200 examples, an annotator is provided with the context, the ground-truth knowledge, model responses and the associated style types. For evaluation of structure style, we defined two kinds of structure styles based on two datasets, namely the Wizard-like style S_{wizard} and the CMU_DoG-like style S_{cmudog}. While for evaluation of content style, we roughly divide content styles in two categories, S_{pos} and S_{neg} for convenience. The responses provided by different models are randomly shuffled to hide their sources. The annotators need to judge the quality of the responses from four aspects: (1) *fluency*: whether the response is fluent without any grammatical errors; (2) *context coherence*: whether the response is coherent with the context; (3) *knowledge relevance*: whether the response is relevant with the knowledge; and (4) *style consistency*: whether the response exhibits the desired style. Each annotator assigns a score from $\{0, 1, 2\}$ (representing "bad", "fair" and "good" respectively) to each response for each aspect. Each response obtains four scores for aforementioned four aspects, and the agreement among all annotators is measured via Fleiss' kappa [2].

Table 4: Human evaluation results on learning structure style.

Models	Wizard Seen					CMU_DoG				
	Fluency	Context Coherence	Knowledge Relevance	Style Consistency	Kappa	Fluency	Context Coherence	Knowledge Relevance	Style Consistency	Kappa
BART	1.68	1.56	1.52	1.34	0.64	1.62	1.57	1.55	1.31	0.63
ZRKGC	1.62	1.59	1.55	1.36	0.65	1.61	1.53	1.65	1.56	0.66
Our	1.71	1.64	1.66	1.77	0.60	1.61	1.66	1.63	1.76	0.74

Results on learning structure style. Table 4 shows human evaluation results on learning structure style. The three models are trained on the Reddit Corpus and then fine-tuned on 10% annotated data. It could be observed that: (1)our model is significantly superior to others on *style consistency*, indicating that the model can learn a consistent expression style with very little data. Specifically, our model tends to directly extract a part of knowledge to synthesize a response after fine-tuned with Wizard data, but learns to generate more knowledge-irrelevant responses in more flexible expression after fine-tuned with CMU_DoG; (2) our model has better performance on *context coherence* and *knowledge relevance*, tallying with its impressive performance in the low-resource scenario.

Table 5: Human evaluation results on learning content style.

Models	Wizard Seen					CMU_DoG				
	Fluency	Context Coherence	Knowledge Relevance	Style Consistency	Kappa	Fluency	Context Coherence	Knowledge Relevance	Style Consistency	Kappa
ECM	0.85	0.94	1.02	1.24	0.65	0.96	0.95	1.18	1.08	0.72
DialoGPT	1.57	1.41	1.19	1.26	0.75	1.55	1.62	1.09	1.02	0.65
Our	1.64	1.60	1.78	1.72	0.76	1.59	1.63	1.51	1.69	0.62

Results on Learning content style. Table 5 reports the human evaluation results on learning content style. The three models are trained on the Reddit Corpus. We can conclude that: (1) by introducing two latent variables and a number of adapters for different styles, our model can generate responses in desired content style (i.e., S_{pos} and S_{neg}) more accurately and achieve significant improvement on *style consistency*, which is consistent with the results in Table **??**; (2) our model also outperforms ECM and DialoGPT on *fluency*,*context coherency* and *knowledge relevance* thanks to the capacity of large-scale pre-trained language models and the introduction of external knowledge respectively.

9 Case Study

Table 6: A case from test set of CMU_DoG.

Knowledge	• MovieName: How to Train Your Dragon • Back at the village, Hiccup subdues a captive dragon in his final training test in front of his father instead of killing it, but Stoick inadvertently angers the dragon into attacking. • Toothless attempts to protect Hiccup in the ensuing panic but is instead captured by the Vikings. • Hiccup accidentally reveals to Stoick that Toothless is capable of locating the dragons' nest. • Stoick disowns his son and sets off for the nest with Toothless chained to the lead ship as a guide. • The Vikings expel most of the dragons but are overwhelmed by the Red Death until Hiccup, Astrid and their fellow pupils fly in riding the training dragons from the academy and provide cover fire. • Hiccup almost drowns trying to break Toothless free from a sinking ship but Stoick saves them both and then reconciles with his son. • Toothless and Hiccup destroy the Red Death but Hiccup is injured in the fight. • Hiccup regains consciousness on Berk where his lower left leg has been replaced by a prosthesis, and the Vikings and the dragons now live in harmony.
Context	User1: Now I keep thinking about phantom pain and whatnot... will he walk with his new prothesis? Was does viking prothesis look like?! User2: ...but now they live there together! Schweet. User1: Yes!
Ground Truth	Probably sharp and imposing!
Wizard Model	They live in a house together . It ' s the same as before but with toothless and hiccup now living on berk where his lower left leg has been replaced by a prosthesis.
CMU_DoG Model	What is the new prothesis ? ! It ' s a Viking
Positive Model	Vikings live in harmony now have the dragons living together and are all still alive to be united.
Negative Model	Though Vikings and the dragons live in harmony, Hiccup can no longer walk with this lower left leg.

This section mainly studies how different models vary in knowledge expression for the same context and background knowledge. Table 6 shows an example from the test set of CMU_DoG. This example contains the background knowledge which gives a plot from the movie, and the dialogue context which is generated by discussing the content in the knowledge. We choose the following four models to generate the response in corresponding style given the dialogue context and knowledge, and all models are pre-trained with the Reddit Corpus: (1) Wizard Model for S_{wizard}: the model fine-tuned with 10% training data in Wizard; (2) CMU_DoG Model for S_{cmudog}: the model fine-tuned with 10% training data in CMU_DoG; (3) Positive Model for S_{pos}: the model forced to express knowledge with positive sentiment; (4) Negative Model for S_{neg}: the model forced to express knowledge with negative sentiment. We can see that the knowlege expression style of the Wizard Model and CMU_DoG Model are quite different. The central part of the Wizard Model response is copied from the background knowledge, which is consistent with the style of Wizard data. The response generated by CMU_DoG Model is more casual in knowledge expression, and the content is mainly related to the conversation context. Besides, responses generated by the Positive Model exhibit evident positive sentiment, while responses generated by the Negative Model show relatively negative sentiment.

References

[1] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR*, 2019.

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

[3] B. Kim, J. Ahn, and G. Kim. Sequential latent knowledge selection for knowledge-grounded dialogue. *arXiv preprint arXiv:2002.07510*, 2020.

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[5] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[6] L. Li, C. Xu, W. Wu, Y. Zhao, X. Zhao, and C. Tao. Zero-resource knowledge-grounded dialogue generation. *arXiv preprint arXiv:2008.12918*, 2020.

[7] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[8] X. Zhao, W. Wu, C. Tao, C. Xu, D. Zhao, and R. Yan. Low-resource knowledge-grounded dialogue generation. *arXiv preprint arXiv:2002.10348*, 2020.

[9] X. Zhao, W. Wu, C. Xu, C. Tao, D. Zhao, and R. Yan. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, 2020.