Knowledge-Grounded Dialogue Generation with Pre-trained Language Models

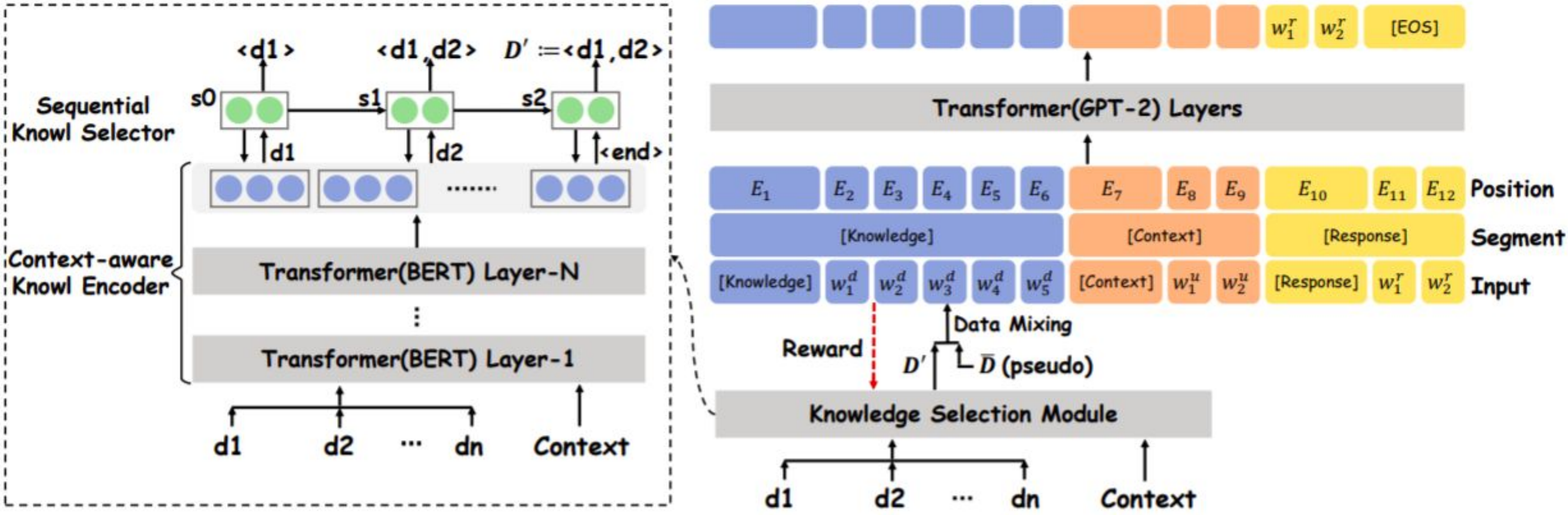
Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, Rui Yan

EMNLP 2020

Task and Method

- Knowledge-aware Dialogue Generation:
  - Input:
    - Conversation Context: The previous sentences in a conversation
    - A Document: Background knowledge that the next sentence in the conversation should be generated with respect to it
  - Output:
    - A text to complete the conversation
- Method:
  - The conversation and the document are fed into GPT-2 to generate the response
- Challenge:
  - The document is long and it will exceeds the GPT-2 limit as the input document
  - Part of the document should be used to represent the knowledge
  - No label is available for knowledge selection from the document

Model



<d1>
<d1,d2>
$D' \coloneqq$ <d1,d2>
Sequential Knowl Selector
s0
s1
s2
d1
d2
<end>
....
Context-aware Knowl Encoder
Transformer(BERT) Layer-N
Transformer(BERT) Layer-1
d1
d2
...
dn
Context
w_1^r
w_2^r
[EOS]
Transformer(GPT-2) Layers
E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10} E_{11} E_{12}
Position
[Knowledge]
[Context]
[Response]
Segment
[Knowledge] w_1^d w_2^d w_3^d w_4^d w_5^d [Context] w_1^u w_2^u [Response] w_1^r w_2^r
Input
Data Mixing
Reward
D'
$\overline{D}$ (pseudo)
Knowledge Selection Module
d1
d2
...
dn
Context

Task

- Dataset: $\{(U_i, D_i, r_i)\}_{i=1}^{N}$
- Goal: $P(r|U, D; \theta)$
- With GPT the goal is defined in autoregressive manner:

$$
\begin{aligned}
P(r|U, D; \theta) &= P(r|g(U, D); \theta) \\
&= \prod_{t=1}^{l_r} P(r_t|g(U, D), r_{1:t-1}; \theta),
\end{aligned}
$$

- Optimize g(U,D) and GPT-2 parameters
- No label for g(U,D)

Knowledge Encoder

- Given $U = (u_1, \ldots, u_n)$ and $D = (d_1, \ldots, d_m)$ construct the sequence $\mathcal{S} = (S_1, \ldots, S_m)$ where (all words of U concatenated with each sentence in D):

$$S_i = [\text{CLS}] w_1^u \ldots w_{l_u}^u [\text{SEP}] w_{i,1}^d \ldots w_{i,j}^d \ldots w_{i,l_d}^d [\text{SEP}]$$

- Using BERT we obtain the representation of CLS for each sentence in the document to have the following sequence of vectors:

$$E = (e_1, \ldots, e_m)$$

Sequential Knowledge Selector

- A sequential process in which a randomly initialized vector is the initial state, using it and the knowledge encoded in E, a sentence is selected and the state is updated until the termination criterion is matched (Max length or Special Token)

$$P(d_i|U, d_{j_{1:t-1}}) = \exp(\alpha_{t,i}) / \sum_i \exp(\alpha_{t,i})$$

$$\alpha_{t,i} = v^\top \tanh(W_e e_i + W_s s_t + b),$$

- At each step the j_t sentence is selected: $\mathrm{argmax}_{i \in \{1,\ldots,m\}} P(d_i|U, d_{j_{1:t-1}})$
- Update state:

$$s_{t+1} = \mathrm{LSTM}(e_{j_t}, s_t)$$

$$U \cup D'$$

Pre-train g(U,D) on Pseudo-Label

- Sort the sentences in D with respect to their similarity to the gold response:

$$\mathrm{Sim}(d_t, r)$$

- Truncate the sorted document in a way that maximizes the similarity between the truncated document and the response:

$$\bar{D} = \{d_{j_1}, \ldots, d_{j_{\bar{m}}}\},$$
$$\bar{m} = \mathrm{argmax}_t(\mathrm{Sim}(d_{j_{1:t}}, r)),$$

- Use the new constructed pseudo-labels to train g(U,D) and GPT-2 using MLE

$$\mathcal{D}_K = \{(U_i, D_i, \bar{D}_i)\}_{i=1}^N$$
$$\mathcal{D}_G = \{(U_i, \bar{D}_i, r_i)\}_{i=1}^N$$

Training g(U,D)

- Use the similarity between response generated by GPT-2 using the document constructed by g(U,D) as the supervision signal
- Increase probability of constructed documents that result in better responses

$$\mathcal{L}_K = -\frac{1}{N}\sum_{i=1}^{N}\left(\tilde{R}_i \sum_{t=1}^{|\tilde{D}_i|} \log P(d_{i,j_t} | U_i, d_{i,j_{1:t-1}})\right)$$

$$\tilde{R}_i = R(\tilde{D}_i) - b,$$

$$R(\tilde{D}_i) = \mathrm{Sim}(r'_i, r_i)$$

$$b = \sum_{i=1}^{N} R(\tilde{D}_i)/N$$

Training GPT-2

- Mix the Pseudo-label and the constructed documents as the input to the GPT-2 and fine-tuned it in an autoregressive manner

$$\mathcal{L}_G = -\frac{1}{N}\sum_{i=1}^{N}\left(z_i \sum_{t=1}^{l_r} \log P(r_{i,t}|U_i, \bar{D}_i, r_{i,1:t-1}) + (1-z_i)\sum_{t=1}^{l_r} \log P(r_{i,t}|U_i, D'_i, r_{i,1:t-1})\right),$$

- Lower the coefficient of loss for the pseudo-label document at each training step

Training Algorithm

Algorithm 1 Optimization Algorithm

```
Input: Training data $\mathcal{D}$, pre-trained GPT-2, initial curriculum rate $p_0$, exponential decay constant $\lambda$, maximum step $M$.
Construct $\mathcal{D}_K$ and $\mathcal{D}_G$.
Optimize $g(U, D)$ and GPT-2 using MLE on $\mathcal{D}_K$ and $\mathcal{D}_G$ respectively.
for $m \leftarrow 1$ to $M$ do
    Sample a mini-batch $\{(U_i, D_i, r_i)\}$ from $\mathcal{D}$.
    Update the parameters of $g(U, D)$ based on Eq.6.    ▷ the Reinforcement Step.
    Sample $\{z_i\}$ from a Bernoulli distribution parameterized by $p$, where $p = p_0 e^{-\lambda m}$.
    Update the parameters of the GPT-2 model based on Eq.7.    ▷ the Curriculum Step.
end for
return $g(U, D)$ and GPT-2.
```

Results

Models	Test Seen					Test Unseen				
	PPL	F1	Average	Extrema	Greedy	PPL	F1	Average	Extrema	Greedy
TMN (Dinan et al., 2019)	66.5	15.9	0.844	0.427	0.658	103.6	14.3	0.839	0.408	0.645
ITDD (Li et al., 2019)	17.8	16.2	0.841	0.425	0.654	44.8	11.4	0.826	0.364	0.624
SKT* (Kim et al., 2020)	52.0	19.3	0.846	0.440	0.665	81.4	16.1	0.839	0.418	0.652
DRD (Zhao et al., 2020)	19.4	19.3	0.852	0.452	0.674	23.0	17.9	0.849	0.439	0.664
SKT+GPT-2*	17.6	20.3	0.866	0.460	0.679	23.7	17.8	0.860	0.437	0.664
GPT-2_{trunc}	14.6(2.2)	18.7(0.7)	0.864(0.002)	0.451(0.006)	0.674(0.004)	16.9(3.1)	18.3(0.6)	0.862(0.002)	0.444(0.005)	0.668(0.003)
KnowledGPT	19.2	**22.0**	**0.872**	**0.463**	**0.682**	22.3	**20.5**	**0.870**	0.452	**0.674**

Thanks